日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: １９９９年１２月２７日

出願番号
Application Number: 平成１１年特許願第３６９９１９号

出願人
Applicant (s): インターナショナル・ビジネス・マシーンズ・コーポレイション

CERTIFIED COPY OF
PRIORITY DOCUMENT

２０００年 ３月１０日

特許庁長官
Commissioner,
Patent Office 近藤隆彦

出証番号 出証特２０００－３０１４４６９

【書類名】　特許願

【整理番号】　JA999282

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 19/00

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　中村　祐一

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　根山　亮

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　戚　乃箴

【発明者】

【住所又は居所】　神奈川県大和市下鶴間１６２３番地１４　日本アイ・ビー・エム株式会社　東京基礎研究所内

【氏名】　梶谷　浩一

【特許出願人】

【識別番号】　390009531

【住所又は居所】　アメリカ合衆国１０５０４、ニューヨーク州アーモンク（番地なし）

【氏名又は名称】　インターナショナル・ビジネス・マシーンズ・コーポレイション

【代理人】

【識別番号】　100086243

【弁理士】

【氏名又は名称】　坂口　博

【選任した代理人】

【識別番号】　100091568

【弁理士】

【氏名又は名称】　市位　嘉宏

【手数料の表示】

【予納台帳番号】　024154

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【包括委任状番号】　9706050

【包括委任状番号】　9704733

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】ワークフロー制御方法、システム、記憶媒体及びサーバ装置

【特許請求の範囲】

【請求項１】

記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された複数の端末装置とを有するシステムにおいて、前記サーバ装置において実行されるワークフローの制御方法であって、

（ａ）いずれかの端末装置からの要求に応じてデータとルールよりなる文書を生成して前記記憶装置に記憶し、

（ｂ）第１の端末装置からの前記文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は前記文書の更新を実行し、

（ｃ）前記文書の処理が終了したかどうかを判定し、終了していないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する

ステップを有するワークフロー制御方法。

【請求項２】

前記ステップ（ｂ）、（ｃ）は、前記サーバ装置において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項１記載の方法。

【請求項３】

前記ステップ（ｂ）はさらに、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連する前記文書のフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知を行うステップを有する請求項１または２記載の方法。

【請求項４】

前記ステップ（ｂ）はさらに、更新要求が妥当でない場合には第１の端末装置に通知するステップを有する請求項１ないし３のいずれか記載の方法。

【請求項５】

前記ステップ（ｂ）はさらに、更新要求を実行する際にタイムアウトとされる

時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行うステップを有する請求項１ないし４のいずれか記載の方法。

【請求項６】

前記ステップ（ｃ）はさらに、処理が異常終了したと判定された場合には、端末装置に通知するステップを有する請求項１ないし５のいずれか記載の方法。

【請求項７】

サーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において実行されるワークフローの制御方法であって、

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、記憶装置上の文書の更新を実行し、

（ｃ）前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する

ステップを有するワークフロー制御方法。

【請求項８】

前記方法は、前記サーバ装置において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項７記載の方法。

【請求項９】

前記ステップ（ｂ）はさらに、更新要求が妥当でない場合には第１の端末装置に通知するステップを有する請求項７または８記載の方法。

【請求項１０】

前記ステップ（ｂ）はさらに、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行うステップを有する請求項７ないし９のいずれか記載の方法。

【請求項１１】

記憶装置とフロー制御部を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記フロー制御部は以下のワークフロー制御機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する

を実行するシステム。

【請求項１２】

前記フロー制御部の機能は、前記フロー制御部において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項１１記載のシステム。

【請求項１３】

前記フロー制御部はさらに前記（ｂ）において、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連する前記文書のフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知を行う機能を有する請求項１１または１２記載のシステム。

【請求項１４】

前記フロー制御部はさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を有する請求項１１ないし１３のいずれか記載のシステム。

【請求項１５】

前記フロー制御部はさらに前記（ｃ）において、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末

装置に所定の通知を行う機能を有する請求項１１ないし１４のいずれか記載のシステム。

【請求項１６】

前記フロー制御部はさらに前記（ｃ）において、処理が異常終了したと判定された場合には、端末装置に通知する機能を有する請求項１１ないし１５のいずれか記載のシステム。

【請求項１７】

記憶装置とフロー制御部を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記フロー制御部は以下のワークフロー制御機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する

を有するシステム。

【請求項１８】

前記フロー制御部の機能は、前記フロー制御部において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項１７記載のシステム。

【請求項１９】

前記フロー制御部はさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を有する請求項１７または１８記載のシステム。

【請求項２０】

前記フロー制御部はさらに前記（ｂ）において、更新要求を実行する際にタイ

ムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行う機能を有する請求項１７ないし１９のいずれか記載のシステム。

【請求項２１】

記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において用いられる、ワークフローを制御するためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、前記プログラムは前記サーバ装置に以下の機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する

を実行させるものである記憶媒体。

【請求項２２】

前記機能は、前記プログラムによって前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項２１記載の記憶媒体。

【請求項２３】

前記プログラムはさらに前記（ｂ）において、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連する前記文書のフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知を行う機能を前記サーバ装置に実行させるものである請求項２１または２２記載の記憶媒体。

【請求項２４】

前記プログラムはさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を前記サーバ装置に実行させるものである請求項

２１ないし２３のいずれか記載の記憶媒体。

【請求項２５】

前記プログラムはさらに前記（ｂ）において、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行う機能を前記サーバ装置に実行させるものである請求項２１ないし２４のいずれか記載の記憶媒体。

【請求項２６】

前記プログラムはさらに前記（ｃ）において、処理が異常終了したと判定された場合には、端末装置に通知する機能を前記サーバ装置に実行させるものである請求項２１ないし２５のいずれか記載の記憶媒体。

【請求項２７】

記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において用いられる、ワークフローを制御するためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、前記プログラムは前記サーバ装置に以下の機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する

を実行させるものである記憶媒体。

【請求項２８】

前記機能は、前記プログラムによって前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項２７記載の記憶媒体。

【請求項２９】

前記プログラムはさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を前記サーバ装置に実行させるものである請求項２７または２８記載の記憶媒体。

【請求項３０】

前記プログラムはさらに前記（ｂ）において、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行う機能を前記サーバ装置に実行させるものである請求項２７ないし２９のいずれか記載の記憶媒体。

【請求項３１】

記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は以下のワークフロー制御機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する

を実行するサーバ装置。

【請求項３２】

前記フロー制御部の機能は、前記フロー制御部において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項３１記載のサーバ装置。

【請求項３３】

前記フロー制御部はさらに前記（ｂ）において、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知を行う機能を有する請求項３１または３２記載のサーバ装置。

【請求項３４】

前記フロー制御部はさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を有する請求項３１ないし３３のいずれか記載のサーバ装置。

【請求項３５】

前記フロー制御部はさらに前記（ｂ）において、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行う機能を有する請求項３１ないし３４のいずれか記載のサーバ装置。

【請求項３６】

前記フロー制御部はさらに前記（ｃ）において、処理が異常終了したと判定された場合には、端末装置に通知する機能を有する請求項３１ないし３５のいずれか記載のサーバ装置。

【請求項３７】

記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は以下のワークフロー制御機能

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、

（ｃ）前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する

を実行するサーバ装置。

【請求項３８】

前記フロー制御部の機能は、前記フロー制御部において前記文書が論理プログラムに変換されてその論理プログラムを実行することにより実現されるものである請求項３７記載のサーバ装置。

【請求項３９】

前記フロー制御部はさらに前記（ｂ）において、更新要求が妥当でない場合には第１の端末装置に通知する機能を有する請求項３７または３８のいずれか記載のサーバ装置。

【請求項４０】

前記フロー制御部はさらに前記（ｂ）において、更新要求を実行する際にタイムアウトとされる時間を登録し、タイムアウトが発生した場合には関連する端末装置に所定の通知を行う機能を有する請求項３７ないし３９のいずれか記載のサーバ装置。

【請求項４１】

記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶する手段と、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行する手段と、、

（ｃ）文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する手段と

を有するサーバ装置。

【請求項４２】

記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は

（ａ）端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶する手段と、

（ｂ）第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行する手段と、

（ｃ）前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求で

ある場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する手段と

を有するサーバ装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】

本発明は、複数の参加者間を文書が流れるようなワークフロー制御システムに関するものであり、特に、その中心的な構成要素であるフロー制御部を対象としたものである。

【０００２】

【従来の技術】

本発明が対象とする典型的な企業間のワークフローの例を図２に示す。まず、顧客が注文要求を出し（２０１）、売り手が商品提供元（サプライヤ）に特定の商品ユニットの予約を要求し、その回答を受け取る（２０２，２０３）。次に、予約が確定したかどうかを顧客に知らせ（２０４）、配送会社に配送の手配をする（２０５）。その後、配送会社は顧客に対して配送日を伝え（２０６）、配送が完了したら売り手に通知する（２０７）。

【０００３】

このようなワークフローに基づいてアプリケーションを開発する場合、従来の方法では、アプリケーション開発者は個々の文書をデータとして定義し、それらが参加者間をどのような順序で文書が流れて行くのかを明示的にプログラムに記述していた。例えば、注文のための文書、カード番号をチェックするための文書、配送依頼の文書などを定義し、これらが顧客、売り手、商品供給元のサプライヤ、配送会社の間を順に流れるといったことを記述的にプログラミングする（例えば、日本アイ・ビー・エム株式会社の製品「ＦｏｒｍＷａｖｅ」（「ＦｏｒｍＷａｖｅ」はInternational Business Machines Corporationの商標））。そして、フローの制御は、このように明示的な文書の流れに従って、参加者への通知や入力情報と制約の整合性のチェックを行いながら、文書の流れを管理することにより行われていた。

【０００４】

文書の流れを明示的に記述する従来の方法では、フローを正確に表現できる反面、記述が煩雑になってしまう場合がある。特に、文書の流れをあらかじめ固定する必要のない場合（または固定的にしたくない場合）には、何種類ものパスが存在し、これらを個別に表現していくことは現実的ではない。例えば、図２では、顧客から始まる一連の流れを固定的に表現しているが、ここでの流れを取り除くと、様々なバリエーションが可能になる。すなわち、最初に配送会社が配送サービスを提供し、それに対して顧客が商品を指定して注文するという全く違った流れも可能である。このことは、フローを固定しないことにより、様々なビジネスプロセスが実現可能になることを示している。しかし、このようなバリエーションに対応するためには、それぞれに対応する記述を準備しなければならずその記述は煩雑となる。

【０００５】

【発明が解決しようとする課題】

本願発明は、以上のような従来技術の問題点に鑑みて、複数の参加者間をビジネス文書が流れるようなワークフロー制御システムについて、新たなフロー制御の手段を提供することを目的とするものである。

【０００６】

さらに、固定的なワークフローに対応するだけでなく、ワークフローの変更に対しても柔軟に対処できるようなワークフローの制御システム等を提供することを目的とする。

【０００７】

また、単に参加者からの入力を受動的に受け付けるだけでは、参加者がいつ入力をしたらよいかが分からないという意味で不十分である。すなわち、参加者が常にポーリングしなければならず、現実的ではない。そのため、文書の状態に応じて、適宜参加者に入力の催促を通知するような機構を提供することも目的とする。

【０００８】

また、要求がキャンセルされたような場合に、それにより生じるフィールドの

リセットを通知して適切な処理が続いて行われるようにすることも目的とする。

【０００９】

【課題を解決するための手段】

本発明はこれらの課題を、記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において実行されるワークフローの制御方法であって、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの前記文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は前記文書の更新を実行し、前記文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知するステップを有するワークフロー制御方法によって解決するものである。

【００１０】

また、本発明はこれらの課題を、サーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において実行されるワークフローの制御方法であって、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、記憶装置上の文書の更新を実行し、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知するステップを有するワークフロー制御方法によって解決するものである。

【００１１】

本発明はこれらの課題を、システムとしては、記憶装置とフロー制御部を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記フロー制御部は、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し

、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知するという機能を実行するシステムによって解決するものである。

【００１２】

また、本発明はこれらの課題を、記憶装置とフロー制御部を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記フロー制御部は、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知するという機能を実行するシステムによって解決するものである。

【００１３】

本発明はこれらの課題を、記憶媒体としては、記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において用いられる、ワークフローを制御するためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、前記プログラムは前記サーバ装置に、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する機能を実行させるものである記憶媒体によって解決するものである。

【００１４】

また、本発明はこれらの課題を、記憶装置を有するサーバ装置と、前記サーバ

装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において用いられる、ワークフローを制御するためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体であって、前記プログラムは前記サーバ装置に、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する機能をを実行させるものである記憶媒体によって解決するものである。

【００１５】

さらに、本発明はこれらの課題を、サーバ装置としては、記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知するワークフロー制御機能を実行するサーバ装置によって解決するものである。

【００１６】

また、本発明はこれらの課題を、記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行し、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールド

をリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知するワークフロー制御機能を実行するサーバ装置によって解決するものである。

【００１７】

また、本発明はこれらの課題を、記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶する手段と、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行する手段と、文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知する手段とを有するサーバ装置によって解決するものである。

【００１８】

また、本発明はこれらの課題を、記憶装置とフロー制御部を有し、ネットワークを介して端末装置と接続されているサーバ装置であって、前記フロー制御部は端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶する手段と、第１の端末装置からの文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は、データベース上の文書の更新を実行する手段と、前記更新要求がキャンセル要求であるか否かを判断し、キャンセル要求である場合は、そのキャンセル要求に関連するフィールドをリセットし、そのリセットされたフィールドに関連する端末装置を同定して通知する手段とを有するサーバ装置によって解決するものである。

【００１９】

【発明の実施の形態】

4.1　発明の概要

本発明は、フローを固定しないようにして記述するための方法として、宣言的な記述を用いる方法を提供するものである。この方法では、後述するように、一連の流れの中で出てくる全てのフィールドを含むような文書を論理的なものとしてデータとルールを含めて定義し、その中でのフィールドに関する制約を記述す

るという方法をとる。このようにすれば、その制約を満たしている限り、参加者はいつでも、どのような変更も可能になり、結果的に様々な流れが可能となり、またワークフローの定義の変更に対しても柔軟に対応できる。

【００２０】

つまり、上述のように従来の記述的なプログラミングによる方法では、文書の流れ（フロー）そのものを明示的にプログラムにより記述しているが、本発明では、文書中のフィールドに対する制約やフィールド間の依存関係などについてデータとルールを含めた宣言的な記述から、論理プログラムに変換して動的にフローを制御するような方法を対象としている。本発明のような宣言的な記述に基づいた方法では、柔軟な記述が可能となる反面、これを現実的なものとするためには、誰に、いつ、何を通知するかを決定する機構が必要不可欠である。以下の実施例では、このような問題を解決するために、以下のような３種類の通知機能を提供する。

1. 必要な時に、入力催促を通知する機能
2. キャンセルによって生じるフィールドのリセットを通知する機能
3. タイムアウトをきっかけとして、入力催促や処理の終了を通知する機能

以上のような機能は論理プログラミング（論理プログラム）の枠組みを利用して実現される。

【００２１】

ここで、論理プログラミングとは、通常の関数プログラミング（例えば、C, Lisp, Pascal など）で中心となる「関数」という概念の変わりに、「関係」という概念を利用してプログラミングする形態のことである。「関数」は、以下のように、与えられた（複数の）入力から出力を導き出すものとして定義される。

F(in1,in2,in3,..) -> out

一方、「関係」では入出力の区別はなく、単に引数が与えられるのみである。

R(p1,p2,p3,...)

（参考までに、関係データベースのテーブルも同様の考え方に基づいている。）

従って、同一の関係を利用する場合でも、与える引数によって違った種類の質問（query）が可能である。例えば、親子関係を表す関係 parent を考えてみると

、親を入力として子供を出力とする質問や、子供を入力として親を出力とする質問が可能になる。

【００２２】

実際の論理プログラミングのプログラミング形態は、関係、引数に使われる変数（?で始まる）、論理演算子（and，or，<-）、を利用して記述される。例えば、子孫関係を表現するルールは以下のようになる。

ancestor(?A, ?D) <- parent(?A, ?D).

ancestor(?A, ?D) <- parent(?A, ?C), ancestor(?C, ?D).

これは、A が D の祖先であるためには、A が D の親であるか、A の子供である C が D の祖先である必要がある、ことを意味している。（本発明におけるルール及び論理プログラムの記法もこれに従う。）

【００２３】

論理プログラムの実行は、ゴールを入力として与えることにより行われる。ゴールとは、関係と引数によって以下のように表現されたものである。

関係名（引数１，引数２，引数３，．．．）

関数の実行と比較すると、ゴールの実行は以下のような特徴がある。

・ゴールを実行すると、そのゴールの真偽値が実行結果として返ってくる。（特定の値は返ってこない）

・引数には、具体的な値だけでなく、変数も指定できる。

・引数に変数が指定してある場合、ゴールが成功したとき（true の時）、変数には具体的な値が代入される。

【００２４】

なお、関数型言語では入力として、関数名（引数１，引数２，．．．）のような情報を与えて実行するが、全ての引数が具体的な値であり、かつ関数の出力として特定の値が返ってくるという点で論理プログラムと異なっている。

【００２５】

本発明では、論理プログラミングの上述の「入出力の区別がない」という特徴も利用している。すなわち、ある参加者からの変更要求が適切かどうかを判断するという観点からポリシーを記述しておき、逆に「どの参加者が何を変更できる

か」という質問をすることにより、次に通知すべき参加者の同定に利用している。

【００２６】

また、これに関連して、論理プログラミングには「後戻り機能」と呼ばれるものもある。これは、ある質問に対する全ての解を求めるためのものである。前述のゴールの実行は成功すると、１組の解を求めることができるが、後戻り機能を利用すると、ゴールの実行過程を強制的に後戻りさせ（１つ前の状態に戻し）、別の可能性も試すことが可能になる。そして、結果的に全ての可能な場合を実行することにより、全ての解が求められることがある。

【００２７】

なお、ここに述べた「関係」という語は「述語」とも呼ばれている。以下では論理プログラミングにおける関係を指す場合としてこの「述語」という言葉も使用する。

【００２８】

図１に本発明のシステム構成を示す。サーバ１１０は記憶装置１１５に接続され、記憶装置にデータベースが記憶される。さらに、サーバはネットワーク１２０を介して端末装置１３０，１４０，１５０，．．．に接続されている。本発明の中心的な構成要素であるフロー制御部はサーバ１１０上に存在し、このフロー制御部によって文書が管理されている。文書は更新のたびに記憶装置上のデータベース１１５に蓄えられる。図ではサーバと記憶装置（データベース）を別の構成としているが、サーバの記憶装置上にデータベースを記憶させるなどしてサーバと記憶装置を一体化することもできる。一方、参加者は端末装置１３０、１４０、１５０・・・を利用してネットワーク１２０を通じてサーバ１１０にアクセスする。ここで、クライアント端末装置は、必ずしもコンピュータ装置に限らず、例えば、固定電話や携帯電話、携帯型の個人端末装置（ＰＤＡ：Ｐｅｒｓｏｎａｌ　Ｄａｔａ　Ａｓｓｉｓｔａｎｔｓ）などを用いることもできる。また、ネットワークとしては、電話回線やインターネット、ＬＡＮ（Ｌｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ）やＷＡＮ（Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒ）などが考えられる。

【００２９】

この図１に示すシステム構成において、本発明がどのように実施されるかを図２に示したワークフローに基づいて以下説明する。ただし、本発明はかかるワークフローに限定されるものではない。顧客が端末装置１３０から注文要求を出すことにより、売り手のサーバ１１０がその注文についての文書（例えば注文票）を生成し、データベース上に記憶させる。そして、サーバが商品提供元（サプライヤ）の端末装置１４０に特定の商品ユニットの予約を要求する。そして、サーバは注文が受け付けられたかどうかを顧客の端末装置１３０に通知し、注文が受け付けられて商品を配送する場合には配送会社の端末装置１５０に配送の手配をする。その後、配送会社は端末装置１５０から顧客の端末装置１３０に対して配送日を通知し、配送が完了したら売り手のサーバ１１０に処理の終了を通知する。

【００３０】

一方、例えば顧客が注文をキャンセルするような場合には、必要な通知が商品提供元や配送会社に通知される。また、商品供給元からの商品の出荷が遅れるような場合にも、そのことを必要な売り手や配送会社などに通知する。但し、このような場合は配送会社の配送日のキャンセルや変更が必要となるので、その必要なフィールドについてのリセットが通知される。さらに、例えば商品供給元が商品の予約が受け入れられるかどうかをなかなか回答しない場合などは、入力を催促する。

【００３１】

そして、本発明は、以下に詳細に説明するように、論理プログラミングの手法を用いて実現した点も重要な特徴である。それにより、従来のワークフローをいちいち記述する記述的なプログラミングに比べて柔軟なワークフローの変更が可能となる。

【００３２】

本発明の中心的な構成要素であるフロー制御部を有するワークフロー制御システム全体の概要を図３に示す。この中では、本発明の中心的な構成要素であるフロー制御部が参加者の端末装置や文書を有するデータベースとどのようなやり取

りをするかが示されている。参加者１は文書に対しての入力要求を端末装置から送信（３１０）し、フロー制御部は入力要求の内容を検証して問題がなければデータベース上の文書を更新する（３２０）。文書が更新された結果に基づいて、フロー制御部は次に「誰が、何をできるのか」を判断し、それに基づいて対象となる参加者に端末装置を介して「通知」を行う（３３０）。本システムは、複数の参加者を対象とできるものであり、他の参加者２も同様の動作が可能となる。

【００３３】

図４にフロー制御部の構成要素を示す。フロー制御部４００は条件判定部４１０と通知部４２０からなる。条件判定部はさらに、参加者からの入力要求を判定する入力チェック機構４１１、ならびに文書に関する処理が終了したのかどうかを判定する終了判定機構４１２からなる。通知部４２０の構成要素としては、入力催促機構４２１、フィールドリセット管理機構４２２、タイムアウト管理機構４２３がある。入力催促機構４２１は、文書の状態に応じて、次に入力できる参加者を同定し、通知する機能を提供する。フィールドリセット管理機構４２２は、文書中のどこかのフィールドがキャンセルされた場合に、それによってリセットされる別のフィールドを同定し、そのフィールドを埋めた参加者に通知する。タイムアウト管理機構４２３は、時間に関連する制約を扱うためのものであり、ある制約のチェックを特定の時間後に行うような機能を提供する。すなわち、制約をタイムアウト時間とともに登録する機能と、その制約のチェックのトリガとなるイベントの生成を行う。なお、これらのフロー制御部の機能は、後述するルールとデータを有する文書を基に論理プログラムを用いて実行することにより実現されるものである。

【００３４】

以下では、個々の構成要素における処理の流れについて詳細に説明していく。まず、説明を具体的にするために図５に示すワークフローを想定することにする。ここでは、顧客からの注文要求（５０１）に応じて、商品提供元に商品ユニットの予約を要求し（５０２）、商品ユニットの予約または拒絶の通知（５０３）に応答して、その旨を通知により顧客に知らせ（５０４）、商品ユニットの顧客への配送に関しては複数の配送会社に競争させることを意図している。つまり、

売り手は、配送の手配を複数の配送会社に対して行い（５０５）、それらの複数の配送会社からのオファーを一定期間（180分）受け付け（５０６）、その後、顧客が配送会社を選択する（５０７）というものである。さらに、配送会社を確定するまでは、顧客は注文のキャンセルが可能であり、配送会社は自分のオファーを取り下げることができるものとする。

【００３５】

4.2 文書の表現

発明を実現するための前提となる文書のデータ構造について説明する。本発明では、文書を単なるデータだけでなく、ルールも含めて論理的にデータとして記述することとしており、以下ではこの文書の構造について説明する。まず、図６に文書の構成を示す。ここにあるように文書は６つの部分から構成されている。それぞれの概要は以下の様になる。

【００３６】

1. Contents（コンテンツ）：１つのワークフローの中で、参照されたり、更新されたりする全てのフィールドを定義したものである。構造は木構造を有し、要素はノードとオプションで値を含む。

【００３７】

2. History（履歴）：Contents中のどのフィールドに対して、誰が、いつ、どのようなオペレーションを実行したかの記録であり、操作のたびに記録を新たに追加する。時刻、人、アクション、ノードＩＤを含む。

【００３８】

3. Access Control（アクセス制御）：フィールドに対するアクセス制御に関する記述であり、ルールとして表現される。対象フィールド、操作する人のロールまたはＩＤ、操作の種類、時間などを用いて規定され、より具体的には図６に示すようにノードＩＤ、タグ名、人、ロール、アクション（操作）及び条件式を含む。アクション（操作）にはw (write), r (read), cr (create), cn (cancel), dl (delete) などがある。

【００３９】

4. Constraints（制約）：Contents 中のフィールドに対する制約の記述である

。通常各フィールド間の関係を記述し、特に時間に関する制約は History 中の記録を用いて記述する。条件式により示される。

【００４０】

5. Dependencies（依存関係）：Contents中の各フィールド間の依存関係の記述であり、「フィールドBはフィールドAに値を書き込んだ後でしか値を書き込めない」といった内容を記述する。Dependenciesは制約の一種であるが、フローの生成やキャンセルが及ぼす影響の同定などに重要な役割を果たすので区別する。被依存ノードＩＤ及び依存ノードＩＤを含む。

【００４１】

6. Termination（終了条件）：ビジネスプロセスの終了条件を記述する。終了には正常終了（Ｅｎｄ）と異常終了（Ａｂｏｒｔ）がある。タイプと条件式を含む。

【００４２】

以上の説明からも分かるように、文書の構成要素のなかで3.のアクセス制御から6.の終了条件は単なるデータではなく、ルールとして記述したものをデータとして蓄えている。これらのルールは後述する判定機構によって解釈され、文書がこれらのルールを満たしているかを検証するために使われる。すなわち、このように文書をデータとルールから記述して、サーバにおいてこの文書を論理プログラムに変換して使用するようにしたことによって、柔軟にワークフローの変更に対処することが可能となる。

【００４３】

文書中の６種類の構成要素の内容と、表現法について具体的に図７に示して詳細に説明する。この図７において、Contents（コンテンツ）中のフィールドOrderID は注文番号、Product は商品に関する情報をまとめたものであり、商品番号（ProductID)、値段（Price)、配送希望日（DeliveryDateRequested）からなる。SupplierID は商品提供元となる業者の番号であり、UnitID は実際の商品ユニット（もの）の番号である。Transport は配送に関する情報であり、複数の配送会社の候補（Candidate）と最終的に決定した配送会社（Specified）からなる。候補には、配送会社番号と配送可能日(DeliveryDateOffered）の情報が含まれる

【００４４】

History（履歴）部の記録は、時刻、操作の主体（人や業者)、操作の種類、操作の対象（フィールド名)からなる。例えば、"14/Sep/1999:15:20:30, Runtime, w, OrderID" は、「時刻 "14/Sep/1999:15:20:30" に、"Runtime" が "OrderID" に対して書き込みを行った」ことを示す。

【００４５】

Access Control（アクセス制御）部において、"value(ConsumerID), w, Specified" は「ConsumerIDに書かれた人は "Specified" に書き込める」ことを示し、"Transport, cr, Candidate#?, (value(Specified) = nil)" は「"Specified" に何も書き込まれていない限り、"Transport" のロールを持った人は "Candidate#?" を作成できる」ことを示す。

【００４６】

Constraints（制約）部において、一番目の制約は「配送会社から提供される配送日（DelvieryDateOffered）は顧客から要求された配送日（DeliveryDateRequested）と同じか前でなければならない」ことを示す。二番目の制約は、制約は時間に関するものであり、「"DeliveryDateRequested" が書き込まれてから180分以内に Specified が書き込まれなければならない」ことを意味し、顧客の注文から180分以内に配送会社の決定をしなければならないこと示す。

【００４７】

Dependencies（依存関係）部では、「"OrderID"、"ConsumerID"の順で書き込む」こと、「"DeliveryDateRequested"、"Candidate#?" の順で書き込む」ことを示している。

【００４８】

Termination（終了条件）部では，ビジネスプロセスが「"Specified" が書き込まれると正常終了」、「"ProductID" がキャンセルした場合と"ProductID" が書き込まれて180分以内に "Specified" が書き込まれない場合にアボート」することが記述されている。

【００４９】

この終了条件部は、正常に終了するための条件と、異常終了の条件が示されている。正常終了の条件としては、TransportSpecified の値が特定された場合が表現されている。一方、異常終了の条件としては、ProductID がキャンセルされた場合、ならびに ProductID が特定されてから、１８０分以内に配送会社のオファーがなかった場合が記述されている。

【００５０】

4.3フロー制御部の実現

4.3.1 処理の概要

図８にフロー制御部の処理の概要を示す。最初に、ある参加者の端末装置からの要求によりデータベース上に図６、図７に示すような文書が生成される（８０１）。文書がデータベース上に生成されると参加者の端末装置からの文書の更新要求を受け付け（８０２）、その更新要求が妥当なものであるか否かを判断する（８０３）。もし妥当でなければ（ｎｏの場合）、そのことを更新要求した参加者に端末装置を介して通知する（８０４）。更新要求が妥当なものである（ｙｅｓ）場合は、データベース上の文書の更新を実行する（８０５）。かかる文書の更新要求の実行後に、文書の処理が終了したかどうかを判定し、正常に終了したか、異常終了したか、まだ終了してないかを判定する（８０６）。異常終了した場合には、参加者に端末装置を通じてそのことを通知して知らせる（８０７）。終了でなければ、次に更新可能な参加者を同定しその端末装置へ通知により知らせる（８０８）。一方、更新要求を実行する際にタイムアウトの登録も行われれるが（この点については後述する）、登録されたタイムアウトが発生した場合には（８０９）登録されている制約を検証し、その後終了判定を行う。

【００５１】

更新要求の処理の流れが、構成要素によってどのように処理されるのかを図９と図１０に示す。ある参加者の要求により生成された文書について、参加者１が更新要求を送信する（図９の９０１）が、更新要求は入力チェック機構へと送られ、入力チェック機構がその更新要求の妥当性を判定する（９０２）。妥当であれば、実際に文書の更新を実行し（９０３）、その後終了条件管理機構へと制御が移り、そこで終了条件の判定をする（９０４）。また、終了でなければ、入力

催促機構に制御が移り（図１０の１００１）、次に入力可能な参加者を同定し、参加者へ入力催促を通知する（１００２）。通知を受けた参加者のうちの一人が次の更新要求を送信し（１００３）、更新要求のチェック、更新要求の実行などを経て、終了条件管理機構が終了判定を下し（１００４）、この文書に関する処理が終了する（１００５）。

【００５２】

4.3.2 文書の内部表現

本明細書では、文書を論理プログラミングの枠組みに基づいて実現する場合についてのべる。これは、論理プログラミングにおける後戻り機能が、本発明を実現する上で重要なものだからである。ただし、本発明はこれに限定するものではなく、当業者であれば容易に理解できるように、後戻り機構を提供する別の推論機構、例えば後ろ向き推論機能を持ったルールベースシステムなども本発明を実現するために利用可能である。以下では、１つの例として論理プログラミングの枠組みに基づいて、文書の内部表現法について詳述する。

【００５３】

（１） コンテンツ

コンテンツはタップル集合として表現される。タップルとは、n 字組みのデータであり、各引数は特定の属性に対応したものである。図１１に示すように、コンテンツは木構造として表現されるが、内部的な表現はこれをテーブル表現したものが用いられる（図１２）。図１１では、各ノードは属性を表現し、葉の部分の楕円で表現されたものは値を表現している。一方、図１２のテーブル表現における各一列がタップルであり、ここでは４引数のタップルとなっている。第一引数はノードの ID を示しており、これは各フィールドにユニークなものをシステムが自動的に割り振る。文書は木構造となっており、ルートからのパスにより、そのノードの位置を表現できるので、第二引数はそのノードのパスを意味し、第三引数でノードの親ノードを示す。第四引数はノード（フィールド）の値であり、nil は何も入ってない状態を意味している。ここでの例では、Consumer 情報が入力された直後の状態を表している。

【００５４】

（２）履歴

図１３は履歴を表現したものとなっており、５引数のタップルとなっている。第一引数は順序、第二引数は時刻を表し、第三引数が操作を行った参加者の IDであり、第四引数が操作を表現している。第五引数がノードの IDである。なお、ここでの記述に関しては、全てのものを記述すると膨大になるので、いくつかのものを抜粋している。

【００５５】

（３）アクセス制御

図１４はアクセス制御を表現したものとなっているがルール表現されている。ルールは以下のような形式で表現されるものであり、条件部が成立すれば結論部が成り立つことを意味している。

結論部 ← 条件部

結論部は３引数の述語として表現されており、access(<node>, <user>, <operation>) は user が node に対して、operation が可能ということを意味している。例えばルール１は、対象ノードのパスが ”/document” であり、ユーザのロールが ”consumer” ならば、書き込み操作が可能（+w）ということを意味している。またルール２は対象ノードのパスが ”/ProductID” であり、ユーザのロールが ”Consumer” であり、そのノードの作成者が対象ユーザ（?User）ならば、書き込み操作が可能（+w）ということを意味している。なお、ルール中において変数は ?XXXX の形式で表現されている。

【００５６】

（４）制約

図１５に示すように、制約は結論部のない条件部のみにより表現される。例えば、制約１は、TransportSpecified の入るCompanyID に関する制約である。制約１の「内容」は特定された配送業者（TransportSpecified）のCompanyIDが所定のMemberに含まれることという制約を受けることを示し、「内容表現」はその実現のための論理プログラムを現している。制約２は DeliveryDateRequested と DeliveryDateOffered の間の関係として表現されている。

【００５７】

（５）依存関係

図１６に示すように、依存関係はフィールド間の関係として表現されており、２引数のタップルとして表現される。意味としては、第１引数のノードの値が特定されているときのみ、第２引数のノードの値をインプットできる、となる。これを実行するためには別途解釈機構が必要であり、それに関しては後述する。

【００５８】

（６）終了条件

終了は、大きく分けて正常終了と異常終了に分類できる。図１７にはさらに３つの終了条件が記述してある。（１）はTransportSpecified が特定されたら、正常終了であることが記述されている。（２）は ProductID がキャンセルされたら異常終了であることが記述されている。また (3) は、ProductID が特定されてから 180 分以内に TransportSpecified を特定する必要があることが記述されている。そして、ここでの述語 timeout は条件のチェックだけでなく、後述する通知のためにも使うことを目的としてこのような表現が取られている。

【００５９】

4.3.3 入力チェック機構

前節での記述を用いながら、更新要求のチェックがどのように行われるかを説明する。例として顧客が商品を特定し、次に商品提供元のサプライヤが Unit ID を特定するような更新要求を出したとする。この場合、入力チェック機構は、アクセス制御、制約、依存関係のそれぞれに関して、その更新要求が妥当か否かを検査する。これらの検査機構はそれぞれ論理プログラミングの枠組みを利用して定義されており、検査機構そのものがルールとして表現されている。アクセス制御に関しては、図１４に示したように、参加者 X がタグ T のフィールドを更新できるかを評価するための述語 allow が以下のように定義されている。

```
allow(?Node, ?Who, 'w') ←
        更新できる条件を記述.
```

そして、allow('/ud:document/ud:contents/Product/UnitID', 'Yamada', 'w') を全てのルールに関して適用し、どれかが true となれば更新可能と判断する。

【００６０】

ここで、ゴールの実行に関して、上記のゴールと図１４のルール１に基づいて具体的に説明する。ゴールの実行は、与えられたゴールとあらかじめ登録されているルールの結論部をマッチングさせ、ルールを具体化することによって行われる。すなわち、具体化された結果、条件部は true となれば、結論部も true となることから、与えられたゴールも true と判断するという考え方である。上記の例では、ゴール allow('/ud:document/ud:contents/Product/UnitID' 、 'Yamada' , 'w') を実行するわけであるが、この場合、図１４のルール１の結論部とゴールとを一致させることにより、ルール自体が次のように具体化される。

allow('/ud:document/ud:contents/Product/UnitID' , 'Yamada' , 'w') ← isPath('/ud:document/ud:contents/Product/UnitID' , "/document") and hasRole('Yamada' , "Consumer").

この中で、ゴールと結論部のマッチングにより、変数 ?Node は '/ud:document/ud:contents/Product/UnitID' となり、?Who が 'Yamada' となるために、条件部も同様に具体化されている。ここで、条件部の isPath はこれらの引数に関して true であり、一方、hasRole もこれらの引数に対して true となるので、結論部も true と判断できる。したがって、ゴールも true となる。

【００６１】

制約に関しては、例として配送会社の Kuroneko がオファー（申し出）する日付として DeliveryDateOffered に 1999-10-08 という値を入れたいという更新要求を出した場合を考える。この場合、フロー制御部はこの要求を「仮に」受け入れて、コンテンツを更新し、新たなコンテンツで制約を満たしているかを検証する。この場合、図１５の制約１を評価すると、DeliveryDateRequested が 1999-10-10 なので、この条件を評価した結果は true となる。成功した場合には、仮のコンテンツを正式に採用し、失敗した場合にはこれを棄却する。

【００６２】

依存関係に関しても、前述と同じ例を考える。この場合には、以下のように、あるノードが依存関係の観点から更新可能かどうかを判断する述語 satisfyDepe

ndency が用意されている。

```
satisfyDependency(?Node) ←
                depend(?Node, ?DependsOn) and
                isFilledAll(?DependsOn).
depend(Node, List). % Node を更新するために必要となるノードの List
...
```

ここで、述語 depend は図１６の表を述語表現したものである。UnitID が更新可能かを判定するには、

```
satisfyDependency('/ud:document/ud:contents/Product/UnitID')
```

を評価すればよく、true であれば更新可能となる。

【００６３】

以上のようにアクセス制御、制約、依存関係の３つを調べることにより、入力チェック機構は更新要求の妥当性を検証できる。

【００６４】

4.3.4 終了条件の判定

終了条件の判定について述べる。正常終了に関しては、endCondition で定義された終了条件を調べるために以下のような述語 checkEnd が用意されている。

```
checkEnd(?Cond) ←
    ?Cond = isFilled(?Tag ) and
    getValue( ?Tag, ?V ) and
    ?V != nill.
```

これは、あるフィールドの値が特定されている場合に終了するような条件の際に使われるものである。CheckEnd は定義されている全ての終了条件に対して呼び出される必要があり、そのための処理のフローは図１８のようになる。正常終了条件リスト（L）について（１８０１）、１つの終了条件を取り出してCheckEndを呼び出して検証する（１８０２）。条件を満たせば(Yes) 終了し、そうでなければ(No) 他の条件を検証する。

【００６５】

また、異常終了に関しては、あるフィールドがキャンセルされた場合と、タイ

ムアウトを判定するために述語 checkAbort が以下のように定義されている。

```
CheckAbort(?Cond)  ←
    ?Cond = isCancelled(?Tag) and
    event(?Time, ?Who, "cn", ?Tag).
```

これは図１７の異常終了の（2）に対応したものである。この場合、 ProductId がキャンセルされた場合にcheckAbort が成功し、異常終了と判定できる。Check Abort も全ての異常終了条件に対して呼び出される必要があり、処理のフローは図１９のようになる。終了条件リスト（L）について（１９０１）、１つの終了条件を取り出してCheckAbortを呼び出して検証する（１９０２）。条件を満たせば(Yes) 終了し、そうでなければ(No) 他の条件を検証する。

【００６６】

4.3.5 入力催促通知機構

この構成要素の役割は、アクセス制御や依存関係に基づいて、誰がどのフィールドを更新できるかを特定し、その参加者に通知することである。他の例においては、全ての引数が具体的な場合について説明しており、この場合はゴールの真偽値によって入力の妥当性を判断することが目的となっているが、この通知機能の実現にあたっては、ゴール(allow) の引数に変数を与えることにより、「誰が」とか「どこを」と言った具体的な値を見つけ出してくることが目的となっている。より具体的には、アクセス制御の観点からは、allow(?Node, ?Who, 'w') を実行すれば、変数にノードと参加者が代入された形で答えを得ることができ、さらに論理プログラミングの後戻り機構を利用して、次々と更新可能なノードと参加者のペアを得ることができる。このことを利用して組み込み述語 findall を利用して、以下のようなゴールを実行するとノードと参加者のペアのリストが List に代入される。

```
findall([?Who, ?Node], allow(?Node, ?Who, 'w'), ?List)
```

この場合、 [?Who, ?Node] のところが変数であり、変数を含むゴールであるので、実行することにより変数に代入される具体的な値が求められる。この場合は、人と対象ノードのペアがリストとして求められる。

同様にして satisfyDependency(N) により、依存関係の観点から更新可能なノードを得ることができる。したがって、allow と satisfyDependency の両方を満たすようなものを求めれば、次に更新通知を発行すべき参加者と、その参加者が更新できるフィールドを求めることができる。 このような要件を満たすルールは以下のように定義できる。

```
canWrite(?X, ?Node) ←
        allow(?Node, ?X, 'w') and
        satisfyDependency(?Node).
```

そして、以下のようなゴールを実行すれば、ノードと参加者のペアを求めることができる。

```
findall([?Node,?Who], canWrite(?Who, ?Node), ?List)
```

例えば、ドキュメントが新規作成された時、このゴールを実行すると、以下のような答えがえられる。

参加者	ノード
'Nakamura'	'/ud:document/ud:contents/Consumer/ConsumerID'
'Seki'	'/ud:document/ud:contents/Consumer/ConsumerID'
'Nakamura'	'/ud:document/ud:contents/Consumer/Name'
'Nakamura'	'/ud:document/ud:contents/Consumer/Phone'
'Seki'	'/ud:document/ud:contents/Consumer/Phone'

【００６７】

このことは、コンシューマーである Nakamura と Seki がともに、ConsumerID, Name, Phone などのフィールドを更新可能であることを意味している。参考までに、述語 findall の処理の概要を図２０に示す。

【００６８】

4.3.6 更新要求の実行の詳細

更新要求の実行の詳細を図２１に示す。最初に、文書への更新の実行し（２１０１）、その後２種類の後処理が続けて行われる。最初に、キャンセル要求かどうかを判定し、yes ならば対象フィールドのキャンセルによってリセットされるフィールドを同定し、そのフィールドをリセットするとともにそのフィールドを

書き込んだ参加者にそのフィールドがリセットされたことを通知する（２１０３）。次に、タイムアウトに関係する制約の登録と削除を実行し（２１０４）、終了条件の判定へ移る。

【００６９】

図２２に更新要求の実行処理において、入力チェック機構とフィールドリセット管理機構がどのように協力するかを示す。更新要求（２２０１）がキャンセル要求として妥当な場合（２２０２）、文書への更新が実行され（２２０３）、その後、キャンセル要求の場合（２２０４）には制御がフィールドリセット管理機構に制御が移る。フィールドリセット機構は関連するフィールドをリセットし、文書を更新する（２２０５）。さらに、参加者にフィールドのリセットを通知する（２２０６）。その後、タイムアウト管理機構へと制御が移るがここでは省略されている。

【００７０】

4.3.7 キャンセルに基づく通知機構

あるフィールドがキャンセルされると、別のフィールドもリセットされる必要がある場合もある。図２２にもあるように、この場合には、関係する参加者（すなわち、キャンセルされるフィールドとそれに依存するフィールドを書き込んだ参加者）にフィールドがリセットされたことを通知する。これは、依存関係に基づいて決定する。例えば、あるノードをキャンセルした時に、誰に通知する必要があるかを調べるには、図２３のように依存関係にあるノードを探し出し、その後各ノードに書き込みをした参加者を同定すれば良い。

【００７１】

図２３では、最初に与えられたノードについて、所定の初期化（２３０１）を行った後に、そのノードに依存するノード集合を見つけ出し（２３０２）、所定の代入を行う（２３０３）。CurrentListが空かどうかを判断し（２３０４）、Noの場合はこれを再帰的に繰り返すことにより、最終的にすべての依存するノードを見つけ出すことができる。CurrentListが空になればAnswerListを得る（２３０５）。

【００７２】

さらに、このようにして見つけられたノードが誰によって書き込みされたかに関しては、履歴情報によって見出すことができる。

【００７３】

処理がある程度進んだ時点でいずれかのノードがキャンセルされた場合に、そのノード自身とそれに依存した他のノードを書き込んだ人に通知が行くことになる。例えば、商品供給元のサプライヤが商品ユニットIDを決め、いくつかの配送会社がビットした後に、顧客が注文をキャンセルしたら、サプライヤと配送会社に通知が行くことになる。この例では、ProductID を入力として、図２３の処理を実行し、さらに履歴からキャンセルを通知すべき参加者を探すと
'Runtime', 'Nakamura', 'FedEx', 'Kuroneko'
を見つけることができる。

【００７４】

これは、ノードProductIDがキャンセルされた時は、'Runtime', 'Nakamura', 'FedEx', 'Kuroneko'（すなわちこのドキュメントに関わっているすべてのユーザ）に、その旨を通知しなければならないことを表している。

【００７５】

4.3.8 タイムアウト管理機構

図２４にタイムアウトの登録と削除の処理の概要を示す。まず、制約の中からタイムアウト述語を含む節を抜き出し（２４０１）、さらに節中のタイムアウト述語の部分を抜き出す（２４０２）。タイムアウト述語は timeout(G1, G2, Interval) のような形式であり、G1 が true となってから、Interval 時間後に G2 が true でなければならないことを意味している。このことから、文書が更新される度に G1 の true/false を調べ、タイムアウトの登録・削除を更新する必要がある。これを反映して、図２４では第一引数の真偽を調べ（２４０３）、真の時は管理テーブルへと追加し（２４０４）、偽のときは管理テーブルから削除する（２４０５）。

【００７６】

図２５にタイムアウト登録・削除とタイムアウトの発生したときの処理を示す

。例えば、図１７の（3）における以下の異常終了条件に関しては、ProductIdが特定された時点で登録され、180分後にタイムアウトが起こることになる。
timeout(isSpecified('ProductID'), isSpecified('TransportSpecified'), 180).

【００７７】

図２５に示すように、タイムアウトの登録と削除（２５０１）のあった後、タイムアウトの発生に応じて（２５０２）、終了条件の判定を行ない（２５０３）、TransportSpecified が特定されていなければ、処理全体が異常終了する。一方、特定されていれば、入力催促機能へと制御が移り、次の催促通知が発行される（２５０４）。

【００７８】

【発明の効果】

本発明により、複数の参加者間をビジネス文書が流れるようなワークフロー制御システムについて、新たなフロー制御の手段を提供することができる。

【００７９】

さらに、固定的なワークフローに対応するだけでなく、ワークフローの変更に対しても柔軟に対処できるようなワークフローの制御システム等を提供することができる。

【００８０】

また、文書の状態に応じて、適宜参加者に入力の催促を通知するような機構を提供することもできる。

【００８１】

また、要求がキャンセルされたような場合に、それにより生じるフィールドのリセットを通知して適切な処理が続いて行われるようにすることもできる。

【図面の簡単な説明】

【図１】

本発明のシステムの構成を示す図である。

【図２】

本発明の対象とする典型的な企業間ワークフローを示す図である。

【図３】

ワークフロー制御システムの概要を示す図である。

【図４】

フロー制御部の構成を示す図である。

【図５】

配送会社からのビットを含んだワークフローを示す図である。

【図６】

文書データの構造を示す図である。

【図７】

文書の例を示す図である。

【図８】

フロー制御部の動作を示す図である。

【図９】

モジュール間の処理の流れを示す図である。

【図１０】

モジュール間の処理の流れを示す図である。

【図１１】

コンテンツの構造を示す図である。

【図１２】

コンテンツの木構造をテーブルとして表現した図である。

【図１３】

履歴の表現例を示す図である。

【図１４】

アクセス制御の表現例を示す図である。

【図１５】

制約の表現例を示す図である。

【図１６】

依存関係の表現例を示す図である。

【図１７】

終了条件の表現例を示す図である。

【図１８】

正常終了の判定を示す図である。

【図１９】

以上終了の判定を示す図である。

【図２０】

全ての要素を見つけるための処理を示す図である。

【図２１】

更新要求の実行の詳細を示す図である。

【図２２】

更新処理の詳細を示す図である。

【図２３】

依存関係にあるノードを見つける処理を示す図である。

【図２４】

タイムアウトの登録と削除の詳細を示す図である。

【図２５】

タイムアウトの登録と発生後の処理を示す図である。

【符号の説明】

１１０　サーバ

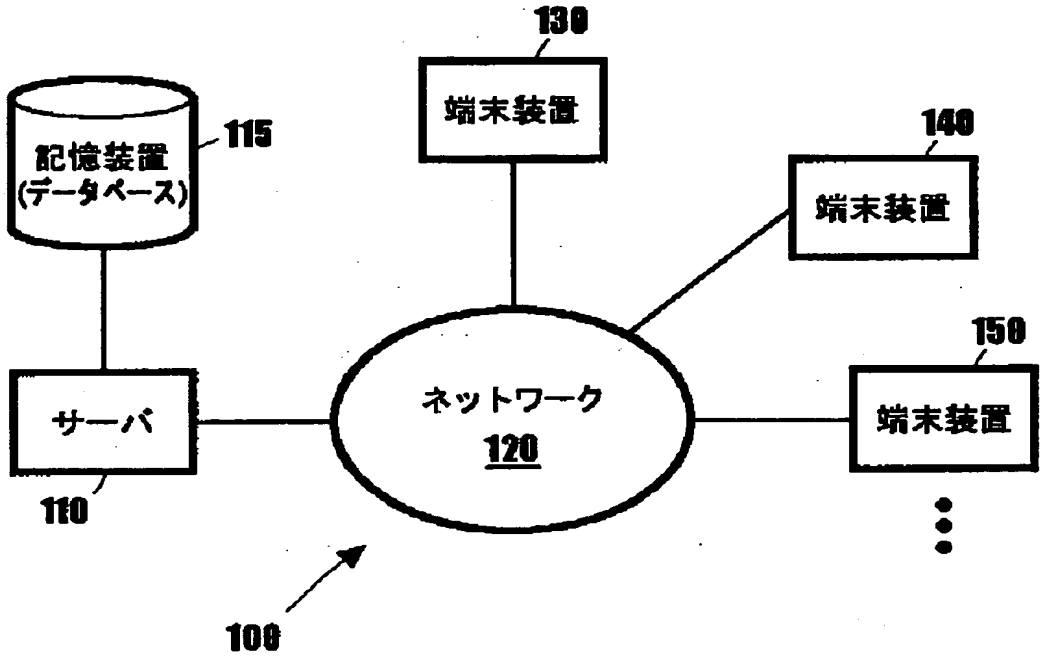
１１５　記憶装置

１２０　ネットワーク

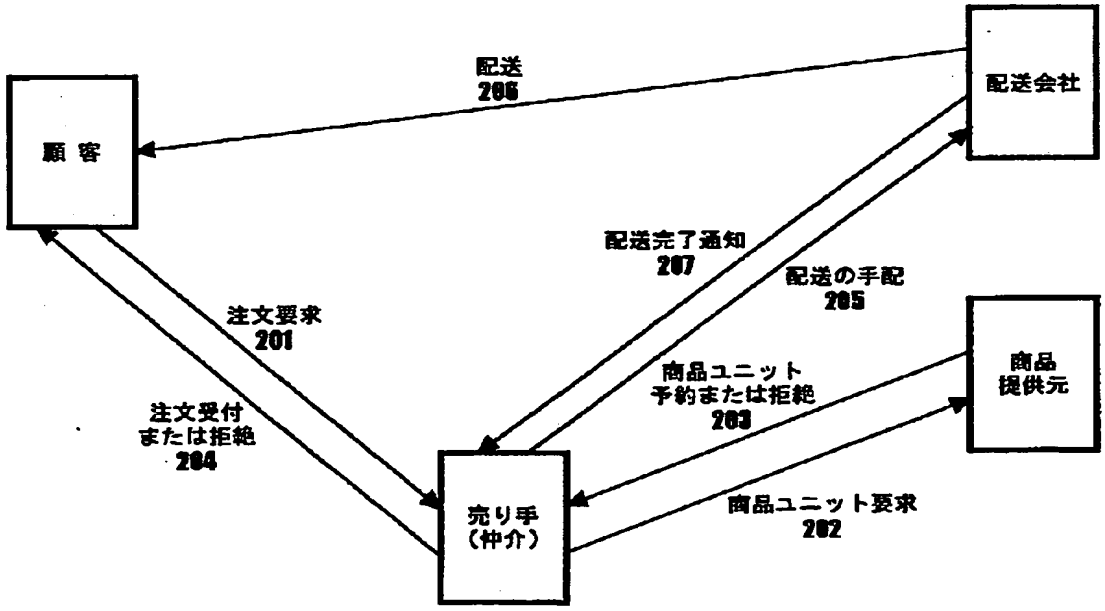
１３０、１４０、１５０　端末装置

【書類名】　　図面

【図１】

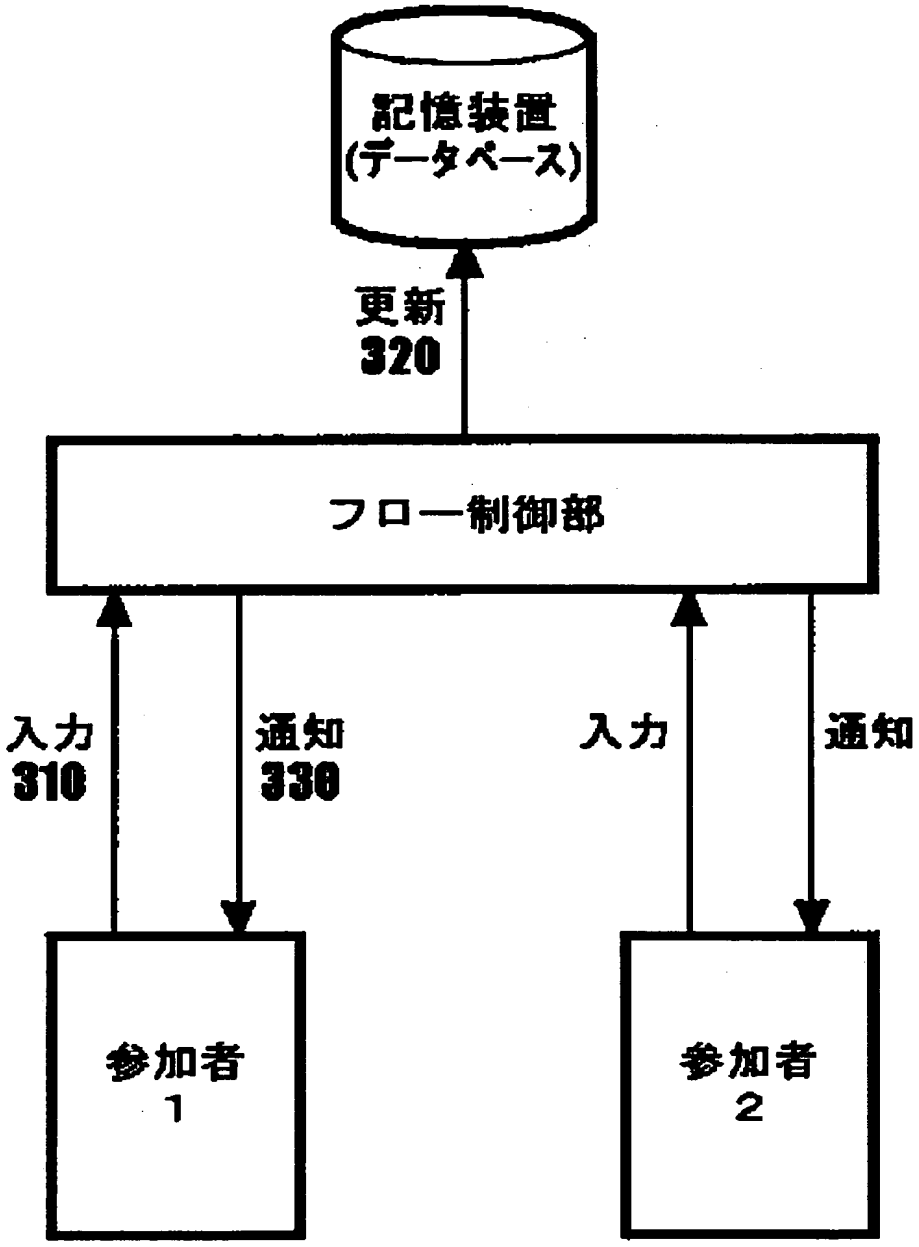


【図２】



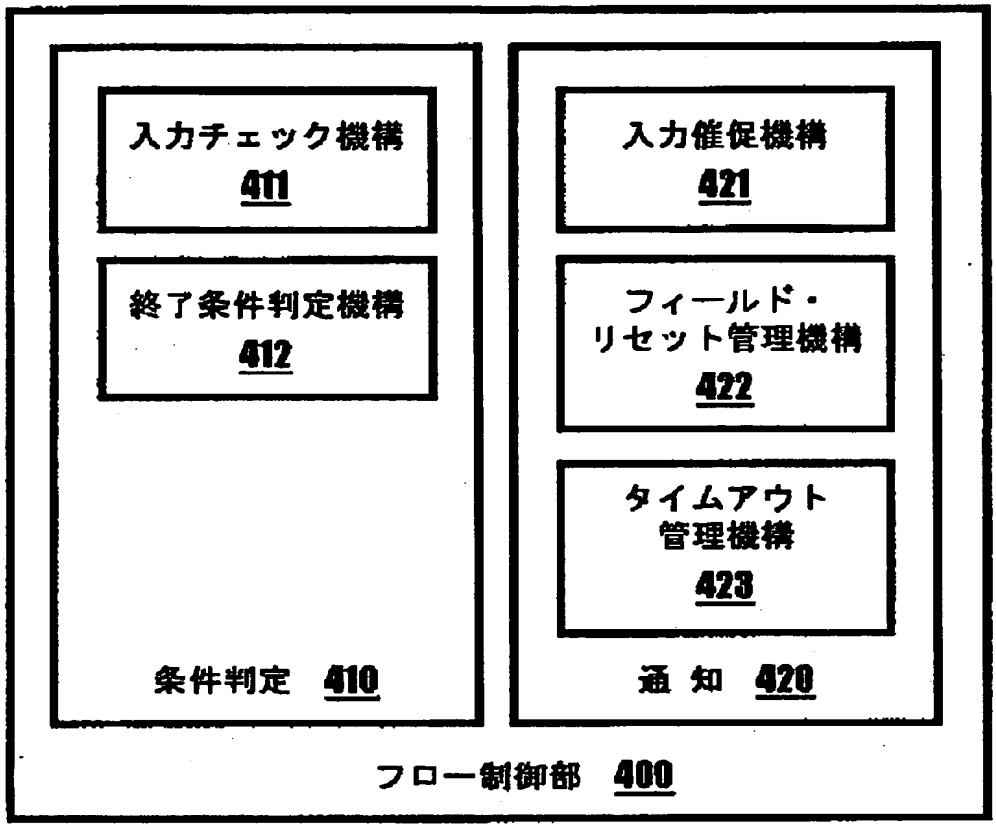
典型的な企業間ワークフロー

【図３】



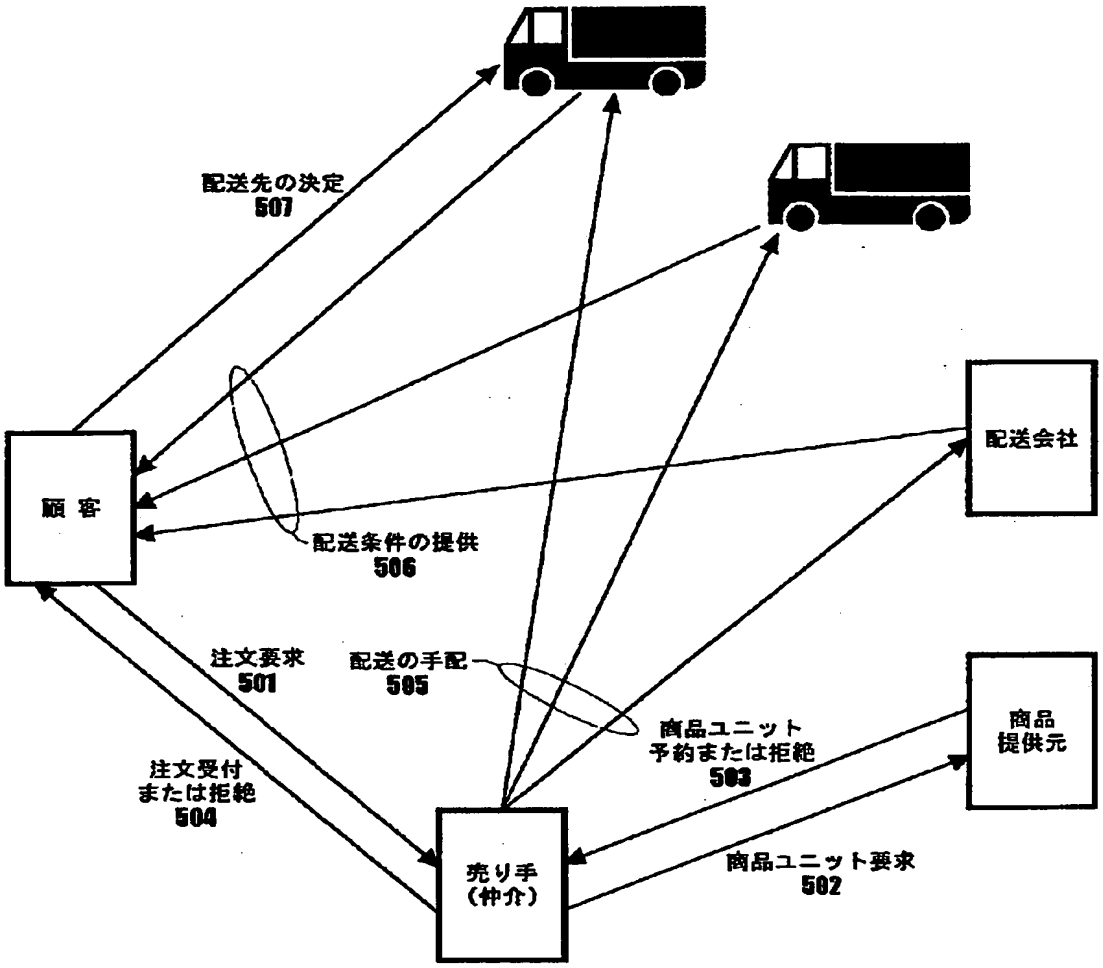
ワークフロー制御システムの概要

【図４】



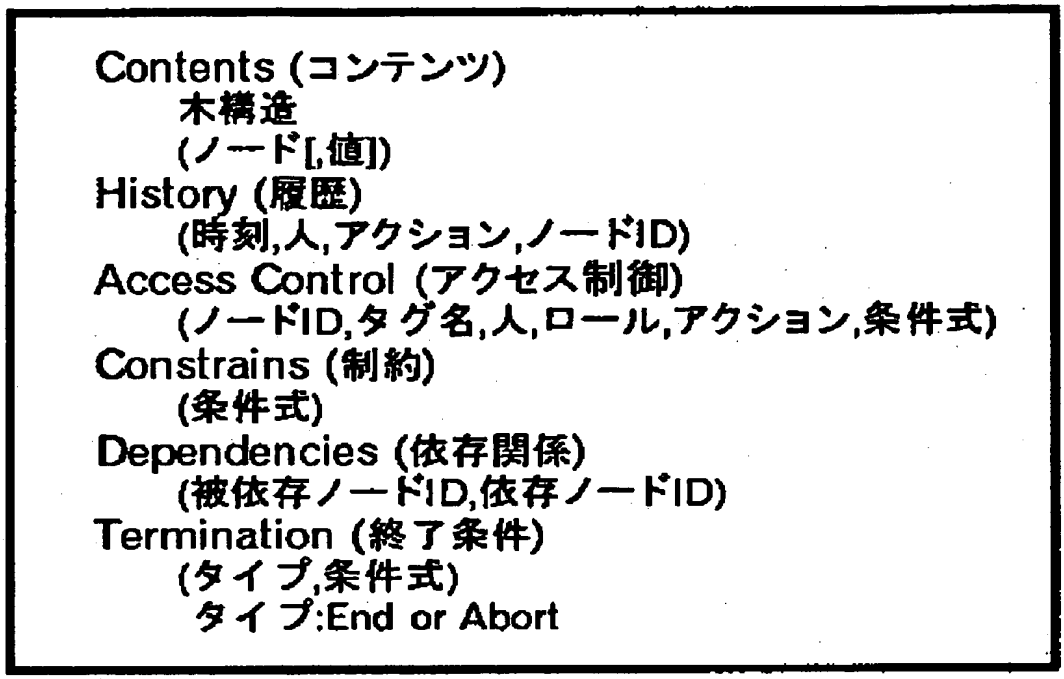
フロー制御部の構成

【図５】



配送会社からのビットを含んだワークフロー

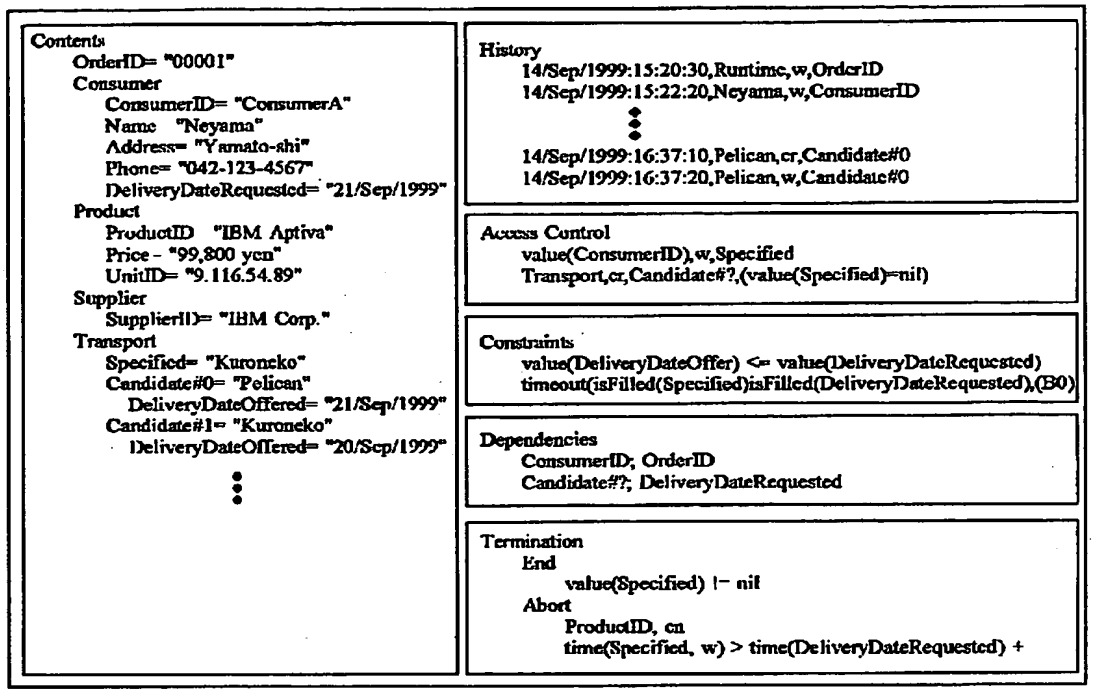
【図６】



[A]はAがオプショナルであることを意味する

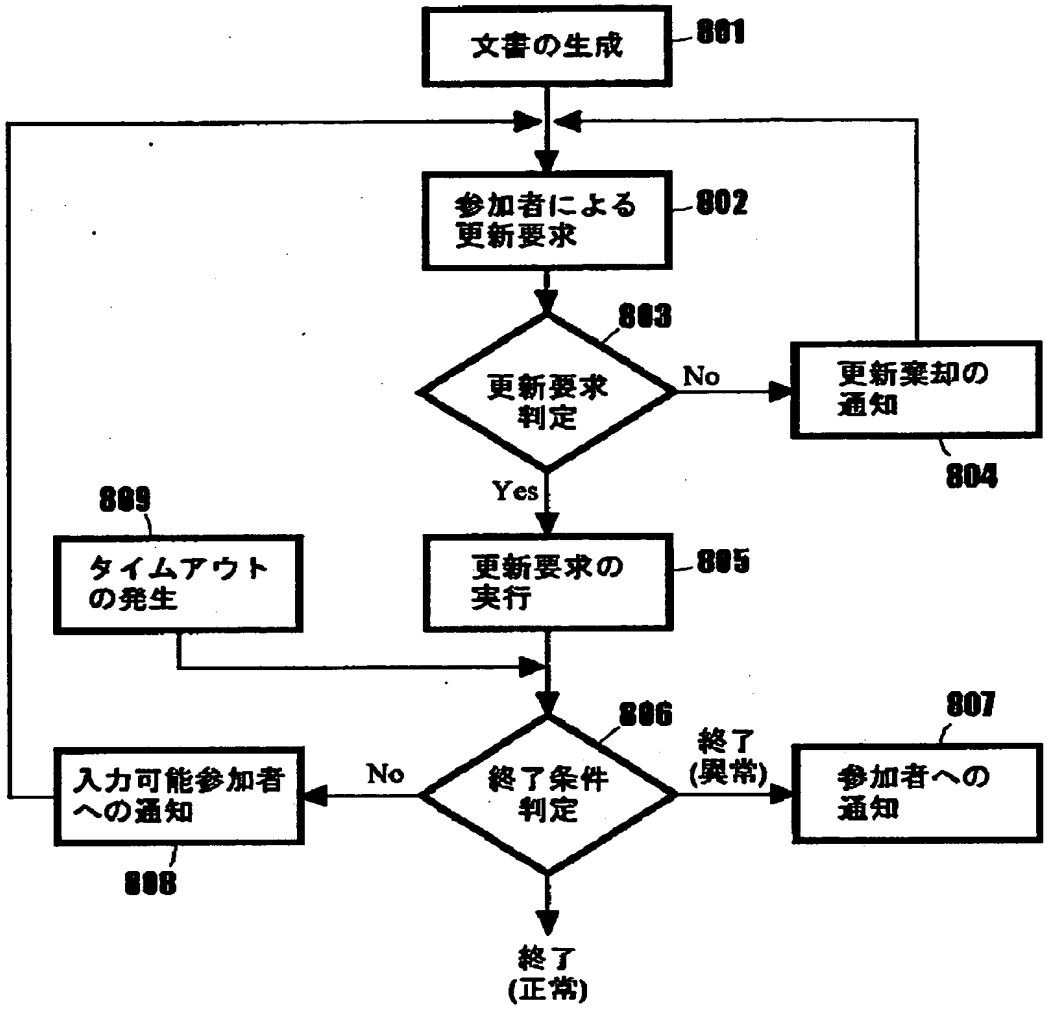
文書データ構造

【図７】


Contents
OrderID= "00001"
Consumer
ConsumerID= "ConsumerA"
Name "Neyama"
Address= "Yamato-shi"
Phone= "042-123-4567"
DeliveryDateRequested= "21/Sep/1999"
Product
ProductID "IBM Aptiva"
Price - "99,800 yen"
UnitID= "9.116.54.89"
Supplier
SupplierID= "IBM Corp."
Transport
Specified= "Kuroneko"
Candidate#0= "Pelican"
DeliveryDateOffered= "21/Sep/1999"
Candidate#1= "Kuroneko"
DeliveryDateOffered= "20/Sep/1999"
History
14/Sep/1999:15:20:30,Runtime,w,OrderID
14/Sep/1999:15:22:20,Neyama,w,ConsumerID
14/Sep/1999:16:37:10,Pelican,cr,Candidate#0
14/Sep/1999:16:37:20,Pelican,w,Candidate#0
Access Control
value(ConsumerID),w,Specified
Transport,cr,Candidate#?,(value(Specified)=nil)
Constraints
value(DeliveryDateOffer) <= value(DeliveryDateRequested)
timeout(isFilled(Specified)isFilled(DeliveryDateRequested),(B0)
Dependencies
ConsumerID; OrderID
Candidate#?; DeliveryDateRequested
Termination
End
value(Specified) != nil
Abort
ProductID, cn
time(Specified, w) > time(DeliveryDateRequested) +

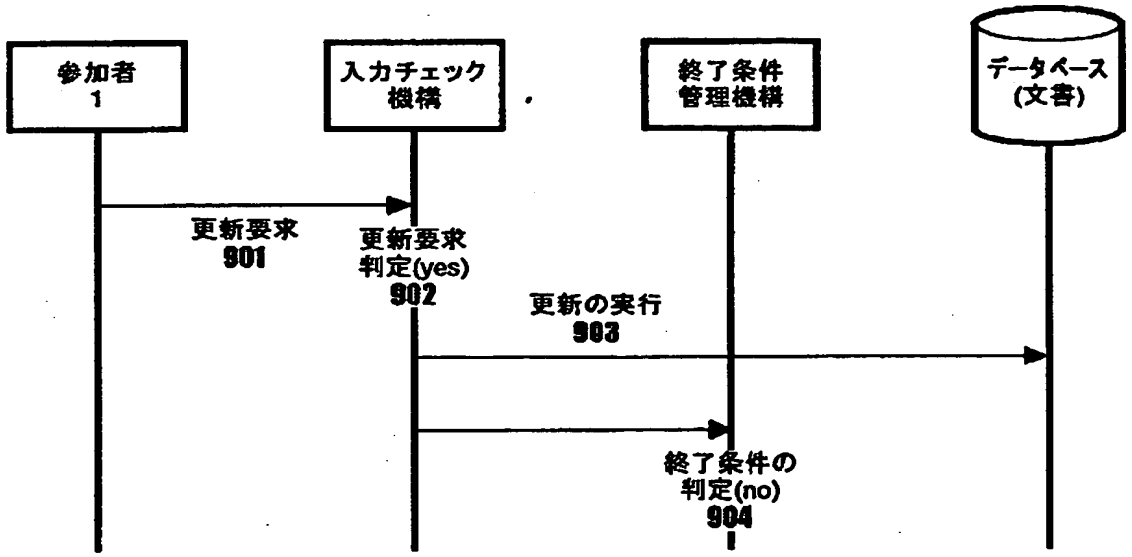
文書の例

【図８】



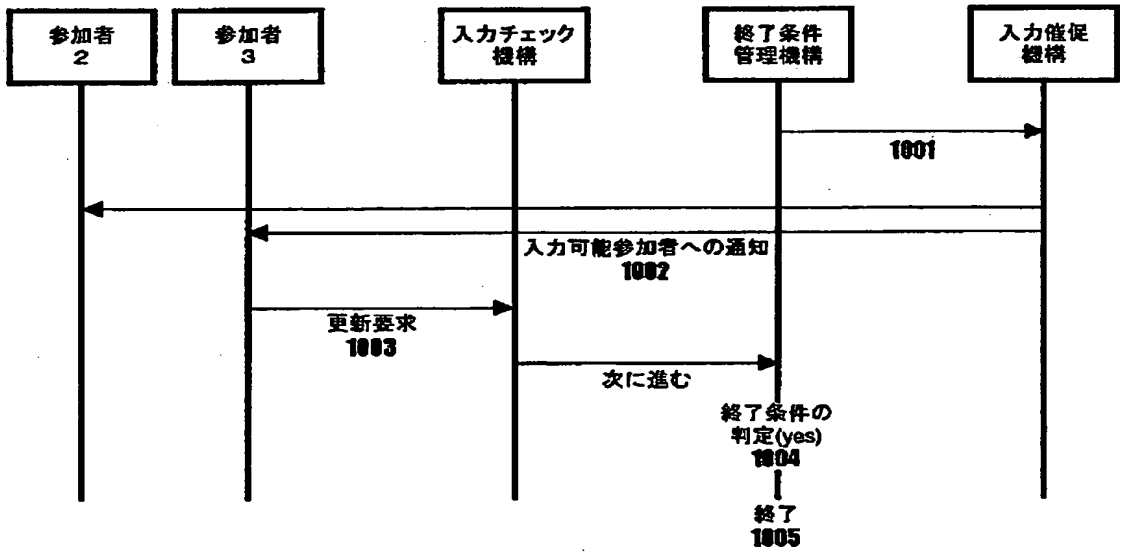
フロー制御部の動作

【図９】



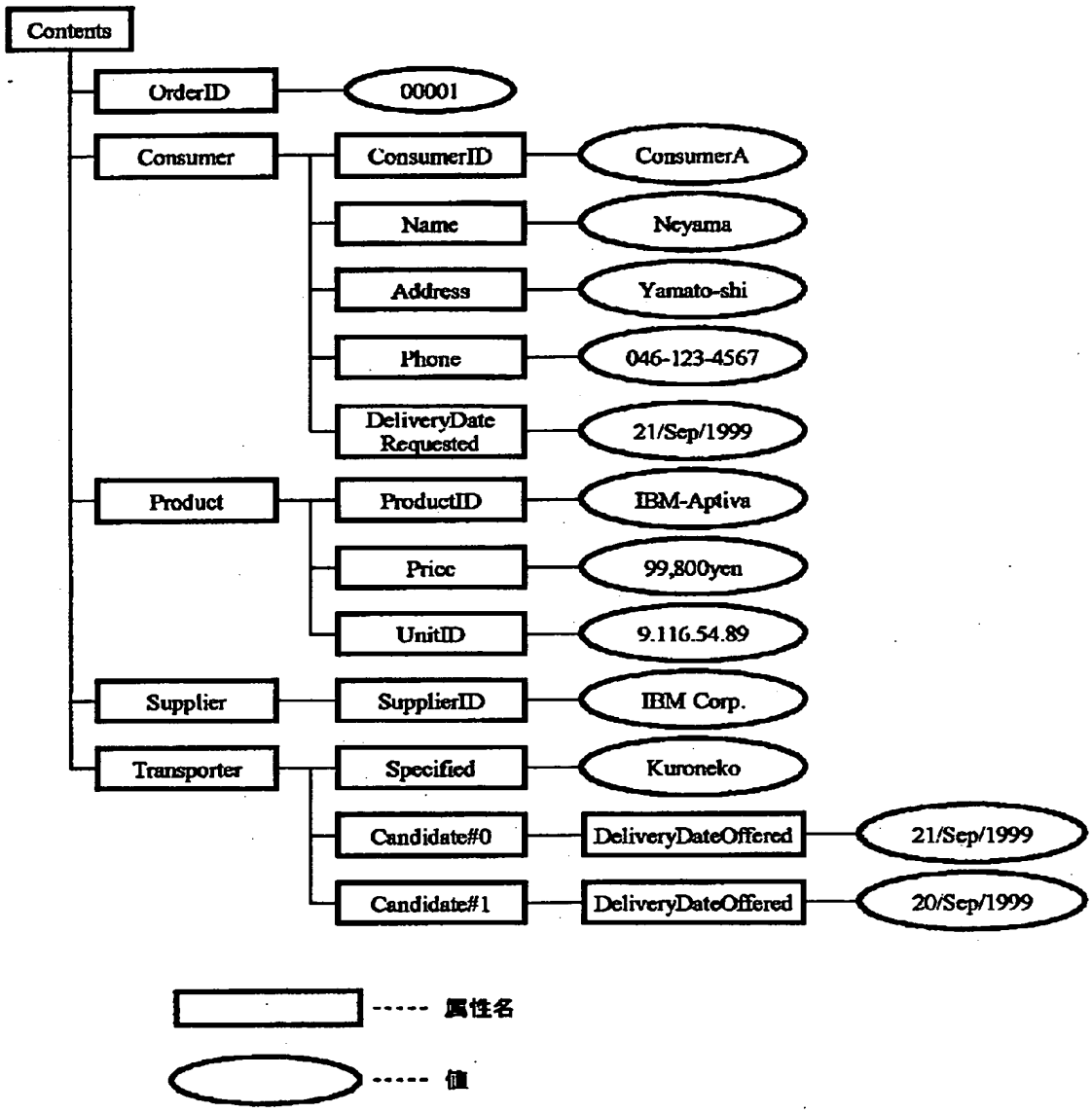
モジュール間の処理の流れ（１）

【図１０】



モジュール間の処理の流れ（２）

【図１１】



コンテンツの構造

【図１２】

	ノードＩＤ（属性名）	親ノードＩＤ（属性名）	値
T0	/	nil	nil
T1	/document	/	nil
T2	/document/contents	/document	nil
T3	/document/contents/OrderID	/document/contents	00001
T4	/document/contents/Consumer	/document/contents	nil
T5	/document/contents/Consumer /ConsumerID	/document/contents /Consumer	Neyama
T6	/document/contents/Consumer /ConsumerID/Name	/document/contents /Consumer	Ryoh Neyama
T7	/document/contents/Consumer /ConsumerID/Address	/document/contents /Consumer	Yamato-shi
T8	/document/contents/Consumer /ConsumerID/Phone	/document/contents /Consumer	046-123-4567

コンテンツの木構造をテーブルとして表現したもの

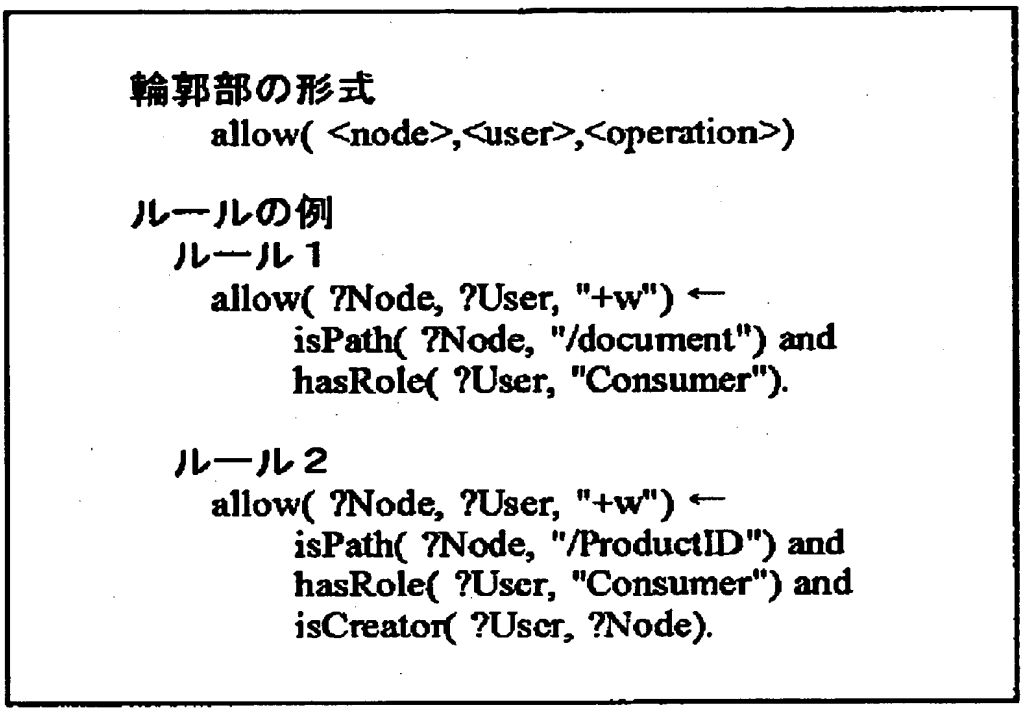
【図１３】

順番	時刻 (sec)	ライターID	アクション	ノードID
0	0	Runtime	Write	/document/contents/OrderID
1	100	Neyama	Write	/document/contents/Consumer /ConsumerID
2	100	Neyama	Write	/document/contents/Consumer /Name
3	100	Neyama	Write	/document/contents/Consumer /Address
4	100	Neyama	Write	/document/contents/Consumer /Phone

(アクションの種類：Create,Write,Read,Cancel)

履歴の表現例

【図１４】



アクセス制御の表現例

【図１５】

制約１
　内容：　member(TransportSpecified, CompanyID)
　内部表現：
　　getValue('TransportsSpecified',V1) and
　　getValueList('CompanyID',V2) and
　　member(V1, V2)

制約２
　内容：　DeliveryDateOffered <= DeliveryDateRequcsted
　内部表現：
　　getValue('DeliveryDateRequcsted',V1) and
　　getValue('DeliveryDateOffered',V2) and
　　V1 <= V2

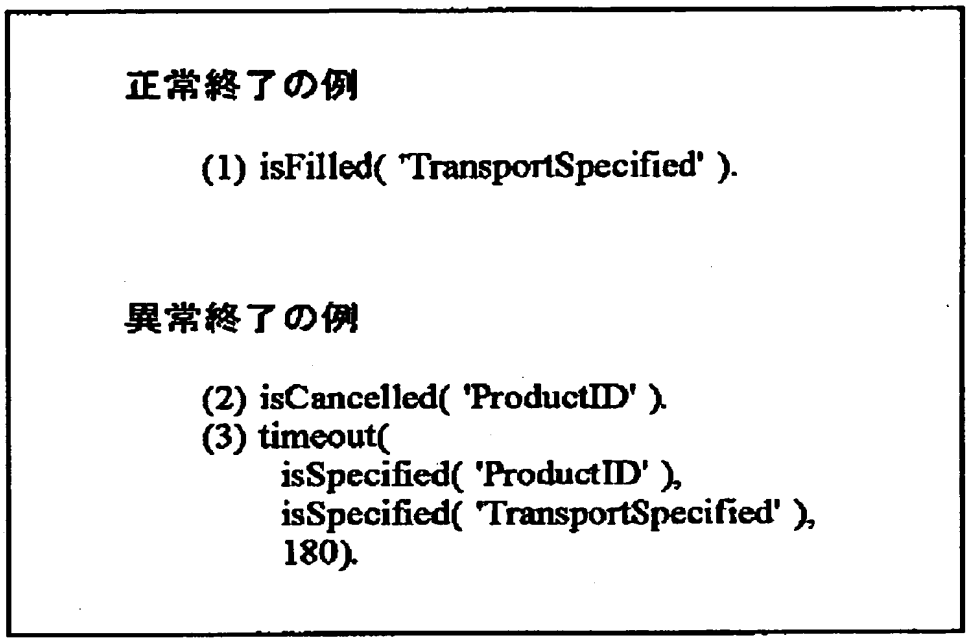
制約の表現例

【図１６】

被依存ノードID	依存ノードID
ProductID	UnitID
UnitID	TransportInfo
TransportInfo	TransportSpecified

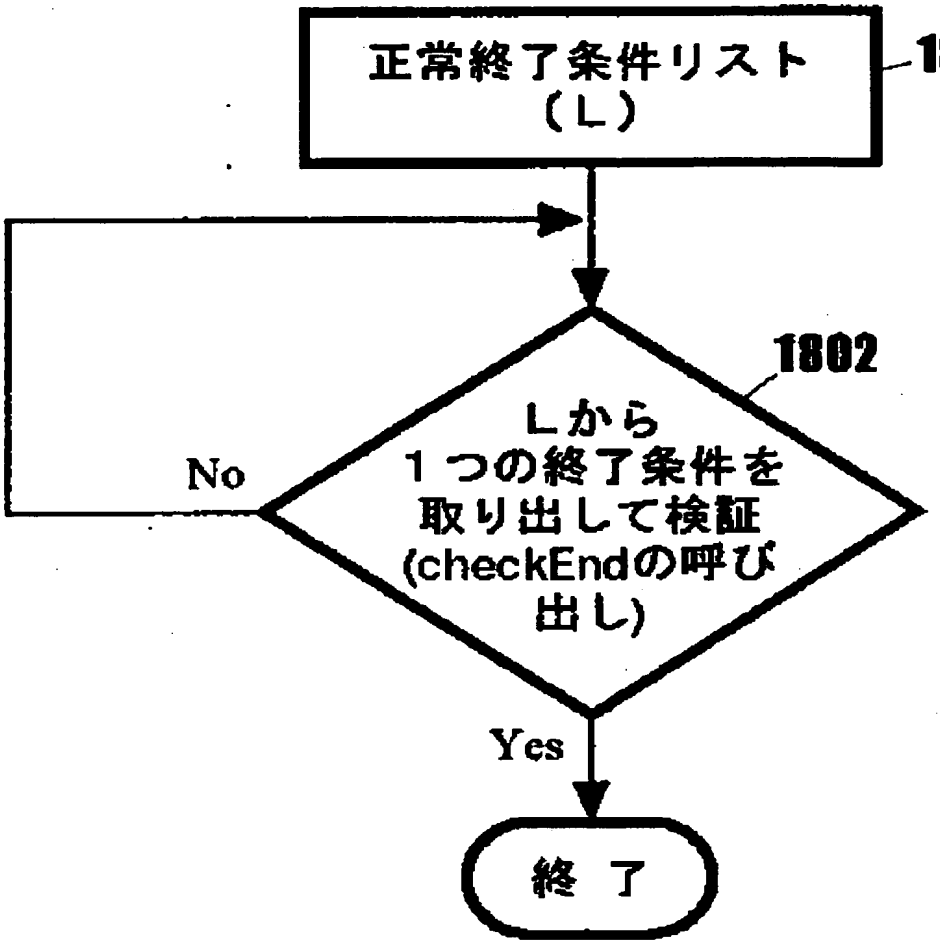
依存関係の表現例

【図１７】



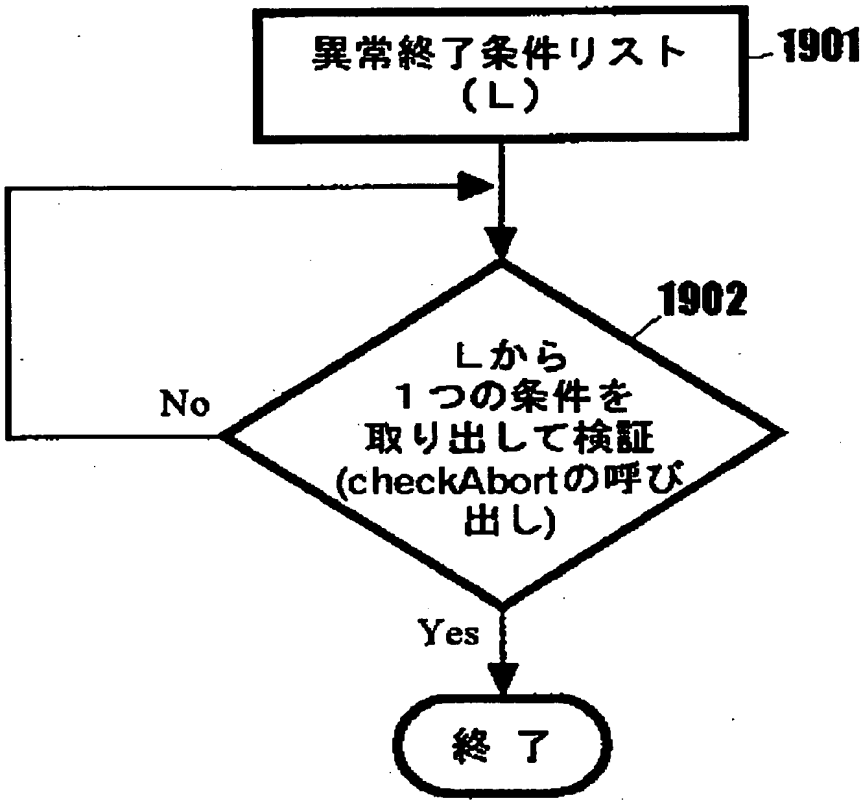
終了条件の表現例

【図１８】


正常終了条件リスト
（L）
1801
1802
Lから
１つの終了条件を
取り出して検証
(checkEndの呼び
出し)
No
Yes
終 了

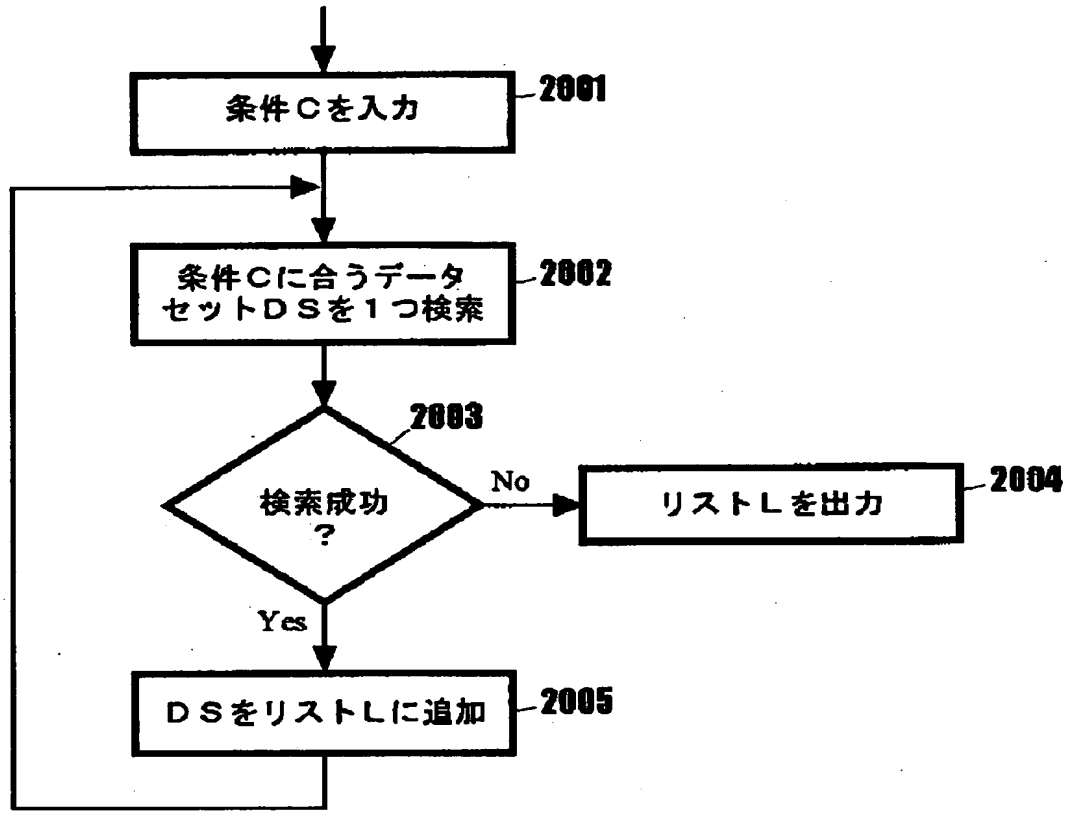
正常終了の判定

【図１９】


異常終了条件リスト
（L）
1901
1902
Lから
１つの条件を
取り出して検証
(checkAbortの呼び
出し)
No
Yes
終 了

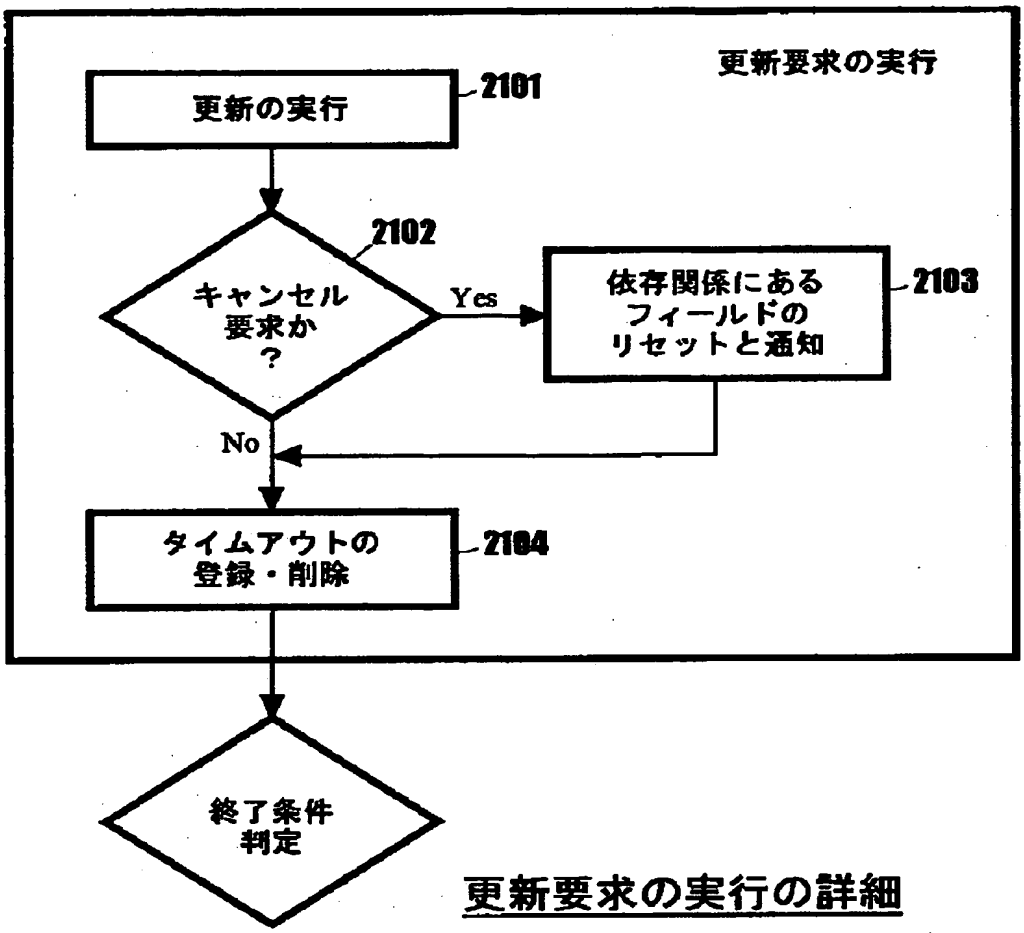
異常終了の判定

【図２０】



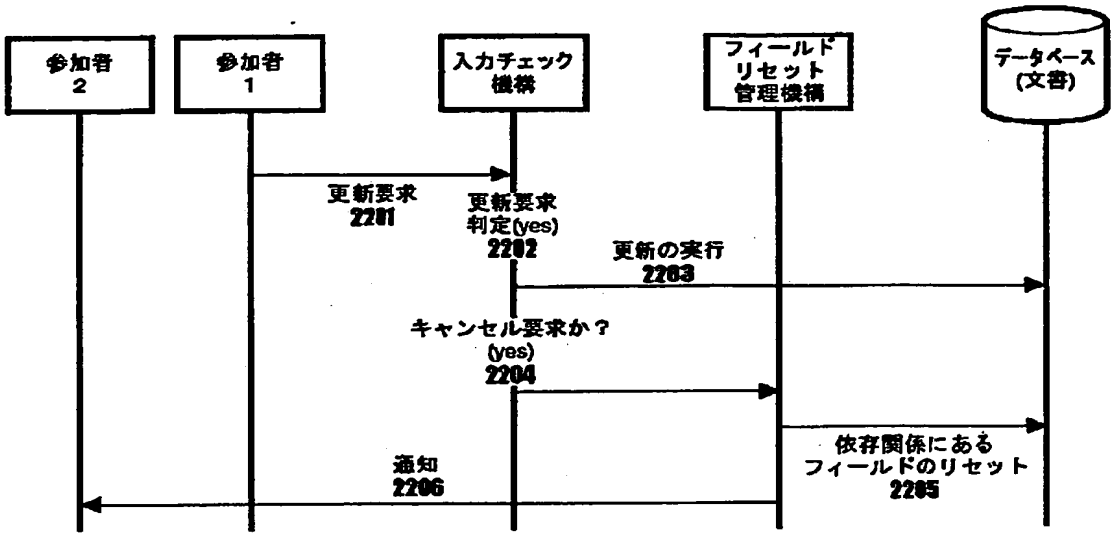
すべての要素を見つけるための処理

【図２１】



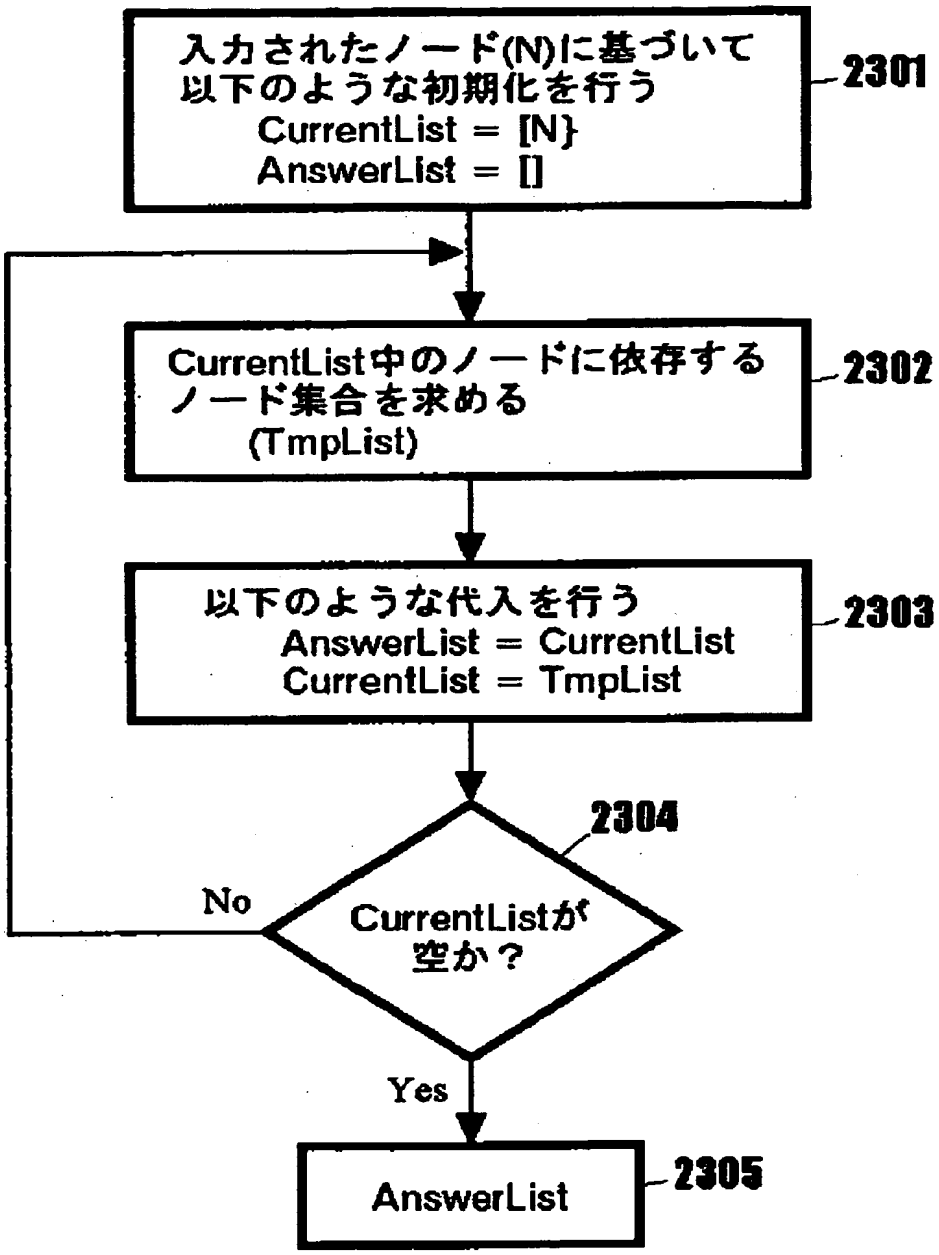
更新要求の実行の詳細

【図２２】


参加者
2
参加者
1
入力チェック
機構
フィールド
リセット
管理機構
データベース
(文書)
更新要求
2201
更新要求
判定(yes)
2202
更新の実行
2203
キャンセル要求か？
(yes)
2204
依存関係にある
フィールドのリセット
2205
通知
2206

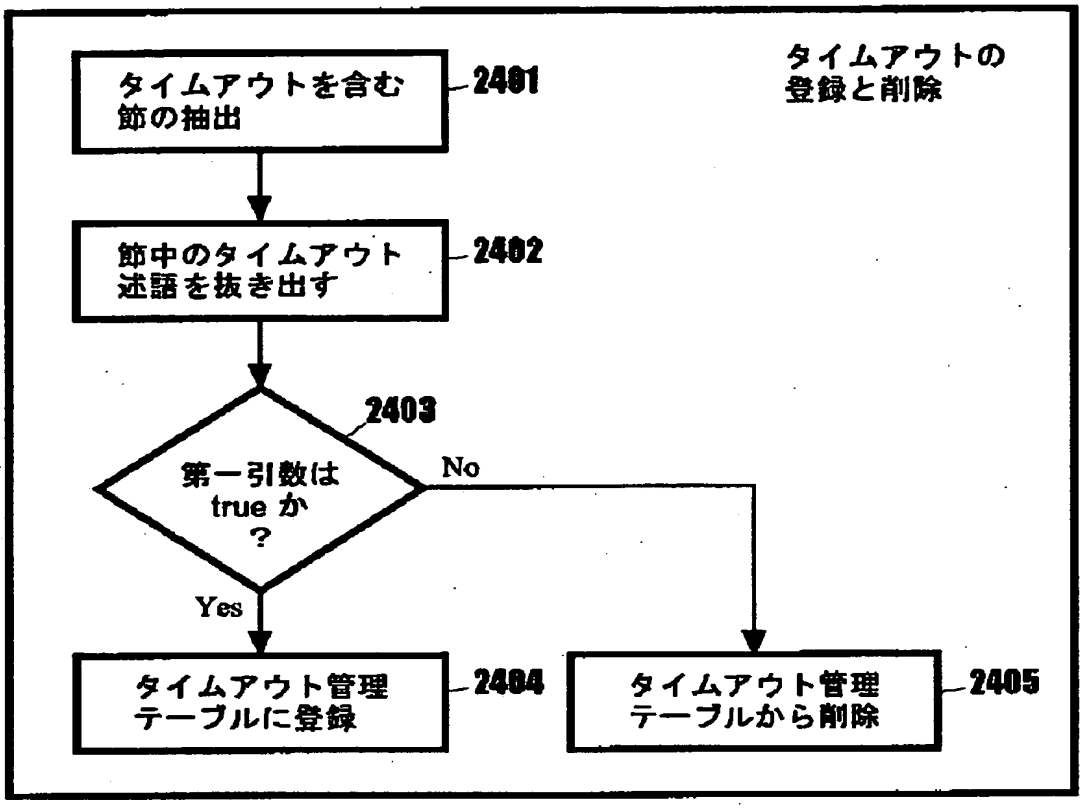
更新処理の詳細

【図２３】



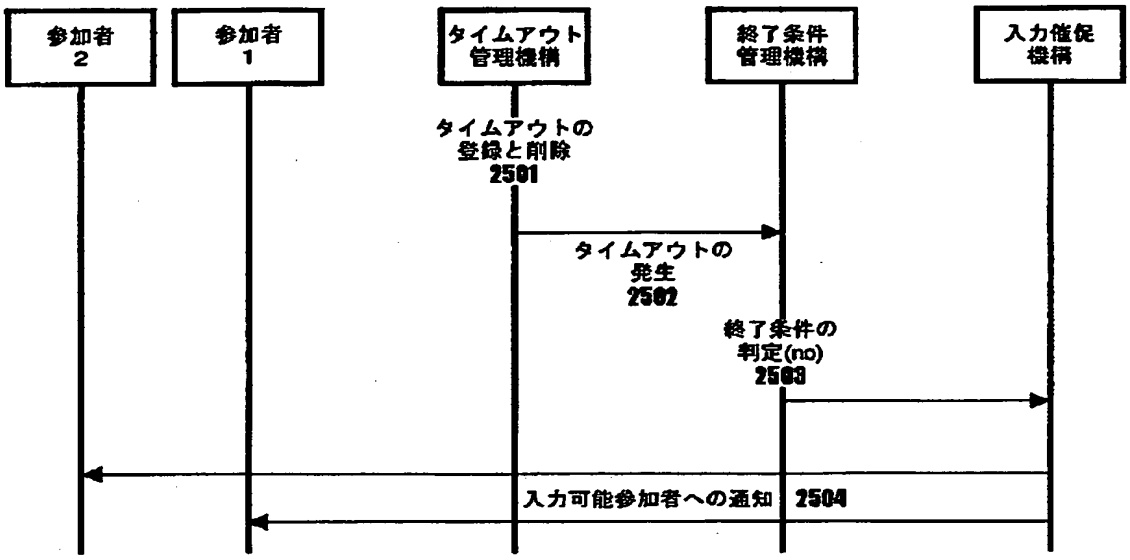
依存関係にあるノードを見つける処理

【図２４】



タイムアウトの登録と削除の詳細

【図２５】



タイムアウトの登録と発生後の処理

【書類名】要約書

【要約】

【課題】

複数の参加者間をビジネス文書が流れるようなワークフロー制御システムについて、新たなフロー制御の手段を提供すること。

【解決手段】

本発明はこれらの課題を、記憶装置を有するサーバ装置と、前記サーバ装置にネットワークを介して接続された端末装置とを有するシステムにおいて、前記サーバ装置において、端末装置からの要求に応じてデータとルールよりなる文書を生成して記憶装置に記憶し、第１の端末装置からの前記文書の更新要求を受け付け、その更新要求が妥当なものであるか否かを判断し、更新要求が妥当なものである場合は前記文書の更新を実行し、前記文書の処理が終了したかどうかを判定し、終了でないと判断された場合には、次に更新可能な第２の端末装置を同定して通知することによって解決するものである。

【選択図】図１

認定・付加情報

特許出願の番号	平成１１年　特許願　第３６９９１９号
受付番号	５９９０１２７１６２７
書類名	特許願
担当官	第七担当上席　　００９６
作成日	平成１２年　１月　４日

＜認定情報・付加情報＞

【提出日】　平成11年12月27日

次頁無

出 願 人 履 歴 情 報

識別番号　〔３９０００９５３１〕

１．変更年月日　１９９０年１０月２４日
［変更理由］　新規登録
住　所　アメリカ合衆国１０５０４、ニューヨーク州　アーモンク　（番地なし）
氏　名　インターナショナル・ビジネス・マシーンズ・コーポレイション